

The Design and Construction of a Hydrogen Line Radio Telescope



Bachelor's Thesis / Project

By: William Barrett Lee

4th October, 2016

Advisers: M.Sc. Michael Gottinger, Prof. Dr.-Ing. Martin Vossiek

In cooperation with:

- The FAU Erlangen Institute of Microwaves and Photonics
- Feuerstein Observatory



Eidesstattliche Erklärung

Ich versichere, dass ich diese Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den

William Barrett Lee

Contents

1	Motivation	3
1.1	Sternwarte Feuerstein - Feuerstein Observatory	3
1.2	Neutral Hydrogen Emissions	4
2	System Overview	7
2.1	System Noise and the Low-Noise Amplifier	8
3	The Antenna	9
3.1	The Parabolic Reflector	11
3.2	Antenna Selection	14
3.3	Antenna Design	15
3.3.1	Helical Antenna Parameters	15
3.3.2	Modeling the Antenna	17
3.3.3	Modeling the Parabolic Reflector	18
3.3.4	Impedance Matching	19
3.3.5	Mechanical Design	21
3.4	Antenna Simulation	22
3.4.1	Antenna and Reflector Simulation	22
3.5	Measurement Results	24
4	Preselect Filter	26
4.1	Overview	27
4.2	Filter Synthesis	28
4.2.1	Define Parameters	28
4.2.2	Low-Pass Prototype	29
4.2.3	Band-Pass Transformation	31
4.2.4	Coupled-Line Transformation	32
4.2.5	Hairpin Filter	36
4.3	Simulation and Optimization	38
4.3.1	Optimization	38
4.3.2	ADS Momentum Simulation	40
4.3.3	CST Frequency Simulation	41
4.4	Measurement Results	43
5	Analog Front-End	45
5.1	Overview	46
5.2	Mechanical Design	48

CONTENTS

5.3	Firmware	49
5.4	Control Software	50
5.5	Measurements	51
6	System Measurements	52
6.1	Putting It All Together	52
6.2	Calibration	53
6.3	Milky Way Crossing	55
7	Summary and Looking Forward	57
7.1	Special Thanks	58
	Appendices	59
A	Analog Front-End Schematics	60
B	Analog Front-End PCB	65
C	Analog Front-End Firmware	70
D	Python Software for Control of the Analog Front-End	73

Chapter 1

Motivation



Figure 1.1: Feuerstein Observatory

1.1 Sternwarte Feuerstein - Feuerstein Observatory

The Feuerstein Observatory was founded in 1999 by Dr. Frank Fleischmann and several others as an astronomy observatory open to the public. It is operated, upgraded, and maintained by a dedicated crew of volunteers who also meet weekly for club meetings. It is located on a remote mountain in the “Franconian Switzerland” region of Bavaria near Ebermannstadt, far from the majority of light pollution from cities.

The observatory has primarily been focused on optical astronomy, but has recently decided, and as a part of this bachelor project, to extend its scope into radio astronomy. They own a 3 meter in diameter parabolic reflector with a motorized mount that is in need of a receiving system for hydrogen-line astronomy.

1.2 Neutral Hydrogen Emissions

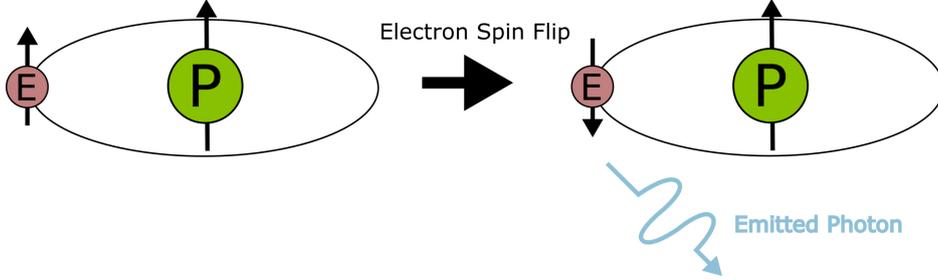


Figure 1.2: Hyper-fine transition of neutral hydrogen

Neutral hydrogen is the simplest and most abundant atom in the universe. It consists of 1 proton and 1 electron, making it electrically neutral. Isolated hydrogen atoms are rare on earth due to its tendency to combine readily with other elements to form compounds. In space, however, large clouds of neutral hydrogen can be found due to its extremely low density and thus ability to avoid combining with other elements or even other hydrogen atoms.

The electron and the proton in a hydrogen atom have spins, the electron- and nuclear-spins respectively, the rotational axes of which are depicted in figure 1.2. The state on the left, showing parallel spins, has a higher energy level than the state on the right, showing anti-parallel spins. The difference between the two energy states is $5.87433 \mu\text{eV}$ [12], which is lost when a photon is radiated at that energy level.

From the well known Planck-Einstein relation, the energy E of an emitted photon is proportional to its frequency f : [9, p.554]

$$E = hf \quad (1.1)$$

Rearranging the equation to solve for the frequency and inserting the energy difference between the two states as ΔE and the Planck constant as h :

$$f = \frac{\Delta E}{h} = \frac{5.87433 \mu\text{eV}}{4.1357 \cdot 10^{-15} \text{ eVs}} = 1.420406 \text{ GHz}$$

With the wavelength, λ being:

$$\lambda = \frac{c}{f} = \frac{2.9979 \cdot 10^8 \frac{\text{m}}{\text{s}}}{1.420406 \text{ GHz}} = 211.06 \text{ mm} \approx 21.1 \text{ cm}$$

Thus, in order to detect the radiation from spaceborne neutral hydrogen,

1.2. NEUTRAL HYDROGEN EMISSIONS

a radio telescope must be developed capable of receiving, amplifying, down-converting, digitizing, and processing a 1.42 GHz electromagnetic wave. Such a radio telescope could be used to measure the doppler shift of the incoming radiation to determine the velocity of the hydrogen clouds relative to earth. This technique is a fundamental measurement in radio astronomy and has been used to map the rotation curve of the milky way, which subsequently can be used to calculate the distance to a certain point within the galaxy. 21.1 cm is also a particularly advantageous wavelength for radio astronomy as the wave is capable of easily penetrating the earth's atmosphere or other sparse matter encountered on the trip to earth.

An example calculation can be seen in figure 1.3. A spike in received radiation is measured with a frequency shift of 8.3 kHz from where we would expect it in a motionless state.

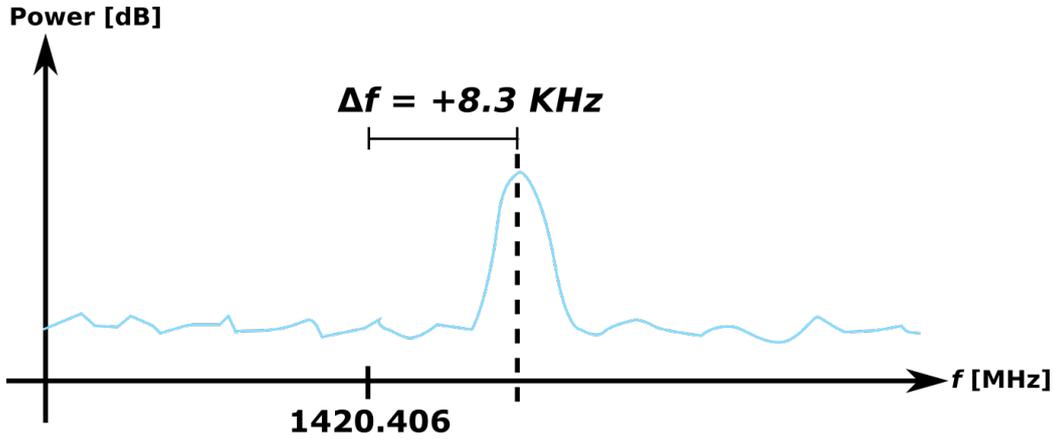


Figure 1.3: Determining relative velocity via doppler shift measurements

The doppler equation is given by: [9, p.345]

$$\Delta f = f_r \frac{v}{c_0}$$

Where Δf is the frequency shift, f_r the rest frequency, v the relative velocity of the object being measured, and c_0 the speed of light. Rearranging for the velocity and inserting the appropriate values:

$$v = \frac{\Delta f}{f_r} c_0 = \frac{+8.3 \text{ kHz}}{1420.406 \text{ MHz}} 300e6 \frac{m}{s} = +1,750 \frac{m}{s}$$

Thus, the measured cloud of neutral hydrogen gas is moving towards us at a relative speed of 1.75 km / s.

1.2. NEUTRAL HYDROGEN EMISSIONS

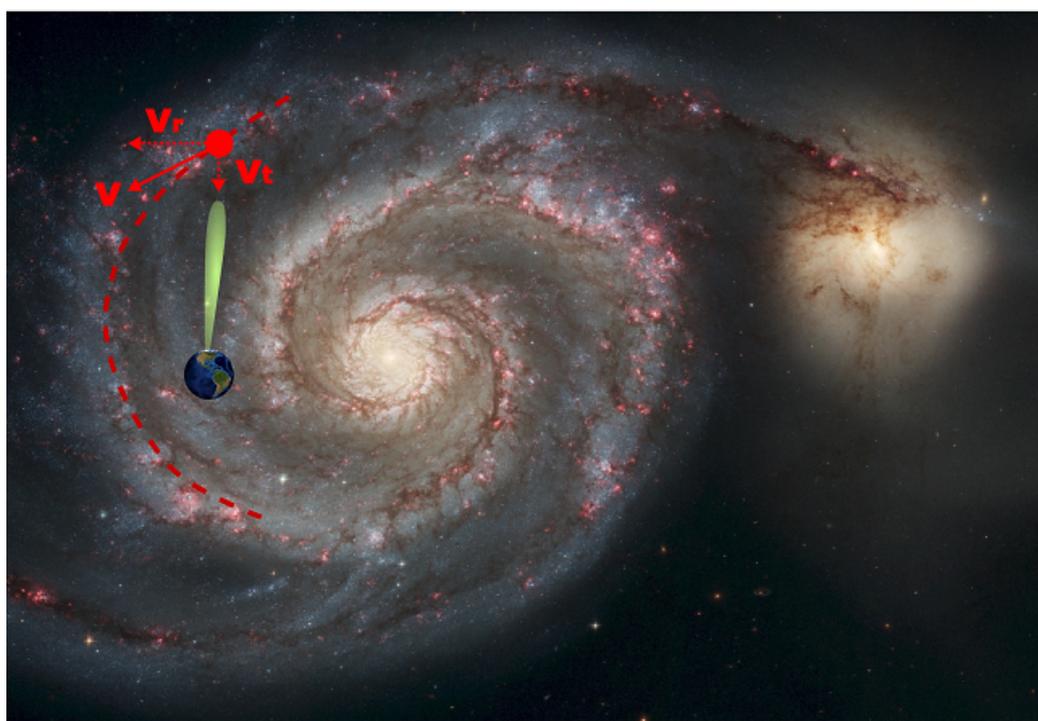
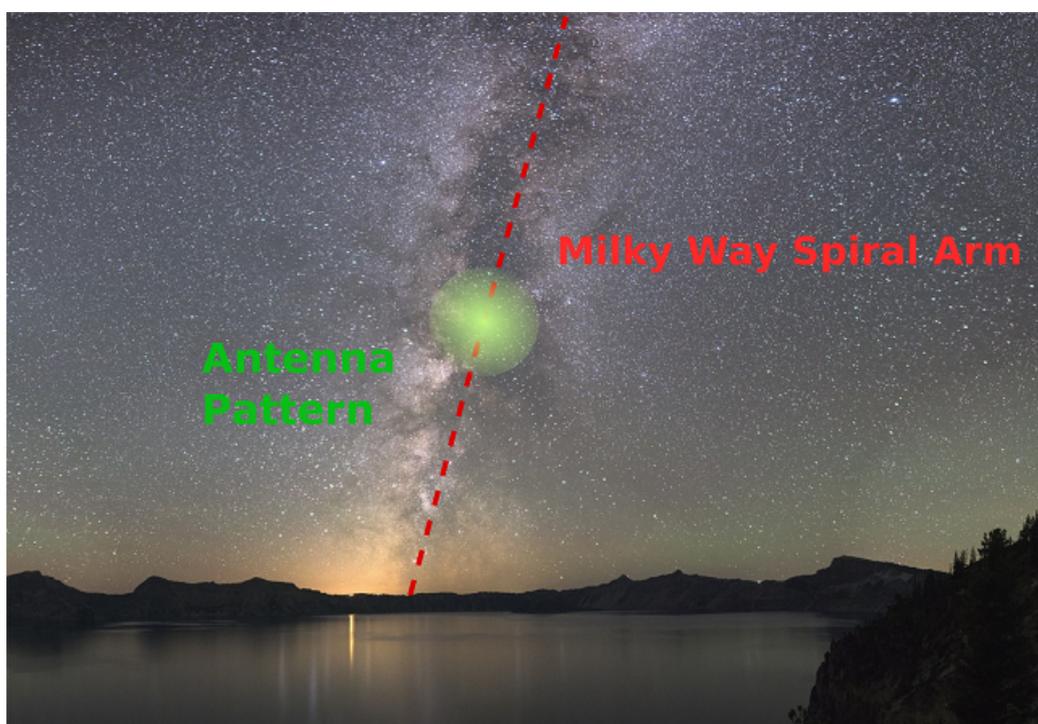


Figure 1.4: Measuring the rotation curve and relative velocity of hydrogen clouds in the Milky Way

Chapter 2

System Overview

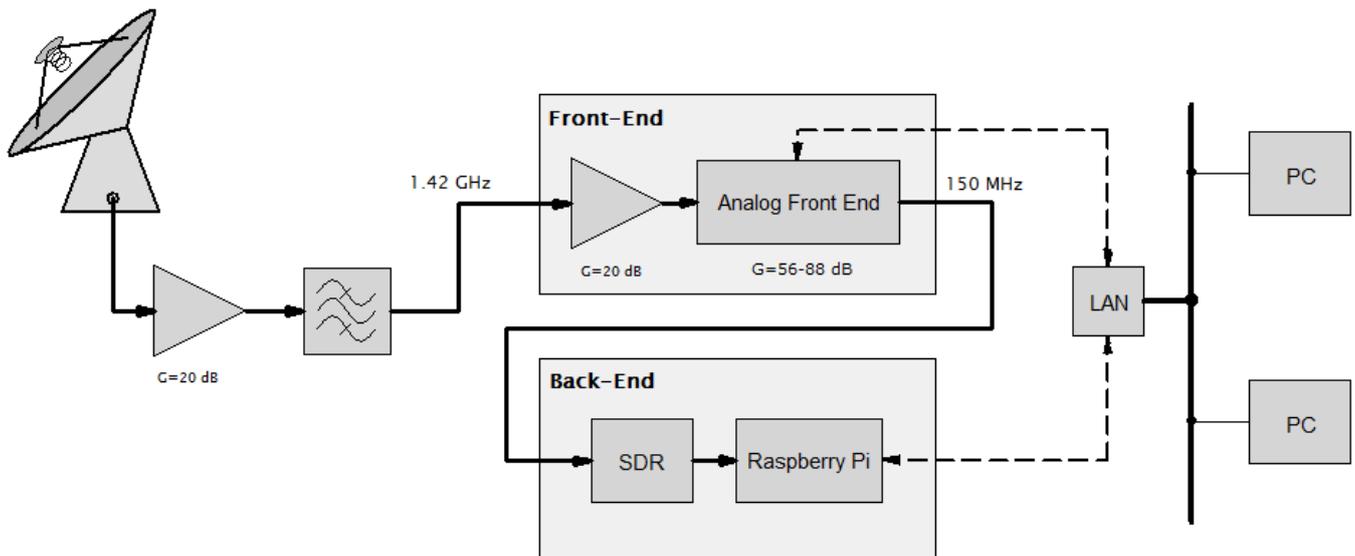


Figure 2.1: System block diagram

The system block diagram is shown in figure 2.1. It was designed to be modular so that individual components could be changed out or upgraded based on need and technological advancements. The signal path begins when the wave front from space hits the parabolic reflector and is redirected towards the custom-made antenna at the focus. It is then converted from an electromagnetic wave into an electrical signal. A very low-noise amplifier with a bandpass preselect filter is mounted directly behind the antenna to give the signal an initial boost to make the 2 meter trip to the analog front-end section.

The analog front-end's task is to downconvert the 1.42 GHz signal to an intermediate frequency of 150 MHz and provide a variable amplification of $76-108$ dB. From there it is sent to the back-end, which will digitize the signal using a software defined radio connected to a Raspberry Pi™, which will send the digitized data over the observatory's local network to a computer for further processing. Both the front- and back-ends can be configured

2.1. SYSTEM NOISE AND THE LOW-NOISE AMPLIFIER

and changed over the LAN interface by any computer with access and sufficient privileges. The total gain of the system is 108-128 dB, meaning a max amplification of 6.3 trillion.

2.1 System Noise and the Low-Noise Amplifier

In order to detect extremely weak signals from space, particular attention needs to be given to the noise figure of the receiving system. The noise figure, defined simply, is the ratio of the system's actual output noise to the level of output noise that would be received if the system were noiseless. Thus, the noise figure needs to be as small as possible.

The noise figure of an entire system, F_T can be calculated with the Friis formula for noise, where F_n and G_n are the noise figures and gains of the individual components: [1, p.139]

$$F_T = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 \cdot G_2} + \dots + \frac{F_n - 1}{G_1 \cdot G_2 \cdot \dots \cdot G_{n-1}} \quad (2.1)$$

A direct realization from this formula is that in a system with cascaded components, the gain and noise-figure of the first component (closest to the antenna) completely dominate the total noise-figure of the system. As such, it is critical to select a low-noise amplifier with the best characteristics possible, i.e., high gain and low noise-figure. For this, a ZX60-P162LN+ from MiniCircuits was chosen, as this would give the rest of the custom-made designs a certain level of tolerance on performance.

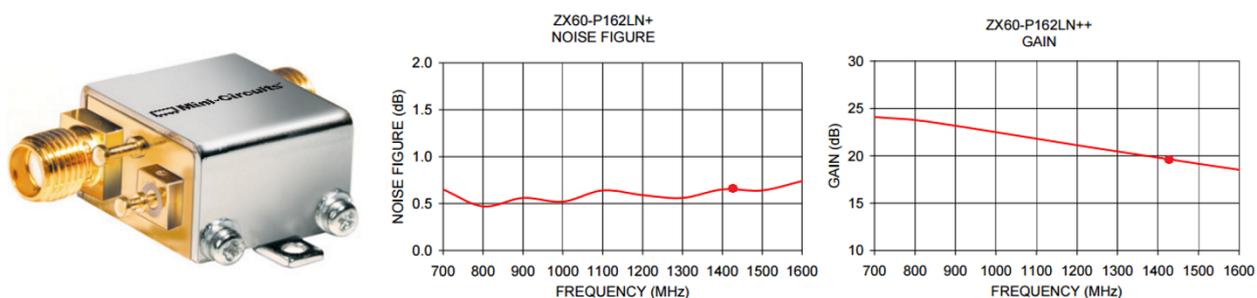


Figure 2.2: The ZX60-P162LN+ (Courtesy of MiniCircuits)

Chapter 3

The Antenna

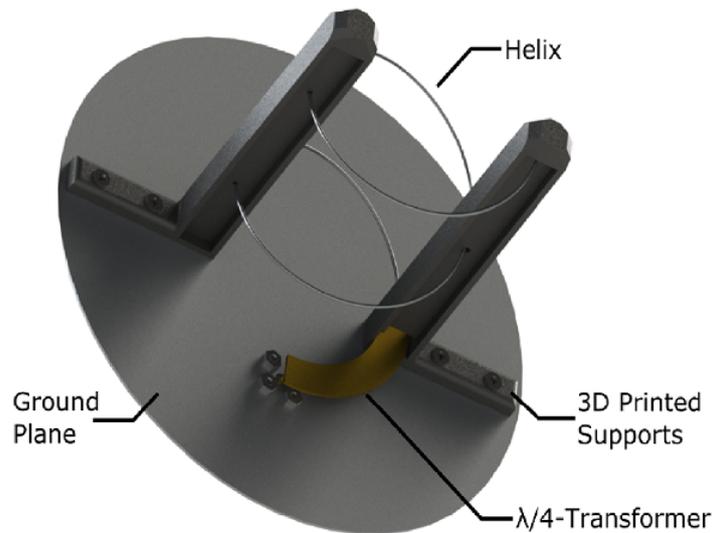


Figure 3.1: The Designed Helical Antenna

3.1 The Parabolic Reflector

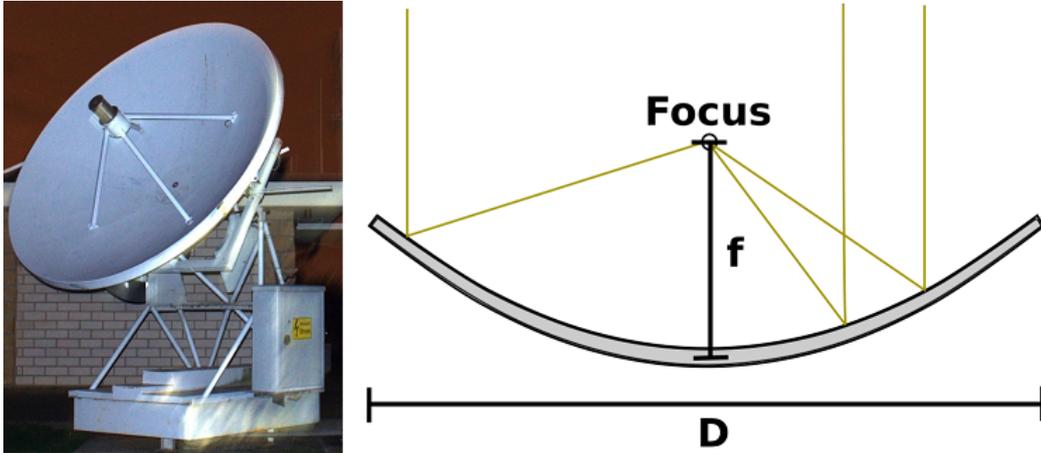


Figure 3.2: Parabolic reflector and its characteristic dimensions

The purpose of the parabolic reflector is to present a large surface area to catch incoming electromagnetic radiation and redirect it to the parabola’s focus, where a receiving antenna will sit and transform the wave into an electrical signal that can be processed and analyzed.

The two defining dimensions of a parabolic reflector are its diameter D , and its focal length, f . The ratio of the focal length to diameter is called the f/D Ratio and is an important piece of information when designing an antenna to fit the reflector.

$$f/D \text{ Ratio: } \frac{f}{D} = \frac{0.89 \text{ m}}{3 \text{ m}} = 0.297$$

The higher the f/D ratio, the further away the focus is from the base of the parabola. An f/D ratio of 0.25 would make the focus flush with the rim of the dish and is the absolute lower limit. This dish, having an f/D ratio of 0.297 is what is called a “Deep Dish” due to how close the focus is to the base. Such a reflector is advantageous for radio astronomy because it limits the amount of radiative energy the antenna sitting at the focus can pick up from the warm ground under the reflector. This concept is illustrated in figure 3.3.

There are three cases of illumination for a reflector: under-, optimal- and over-illumination. In the case of under-illumination, the gain of the feed antenna is too high for the reflector, which causes certain parts of the dish

3.1. THE PARABOLIC REFLECTOR

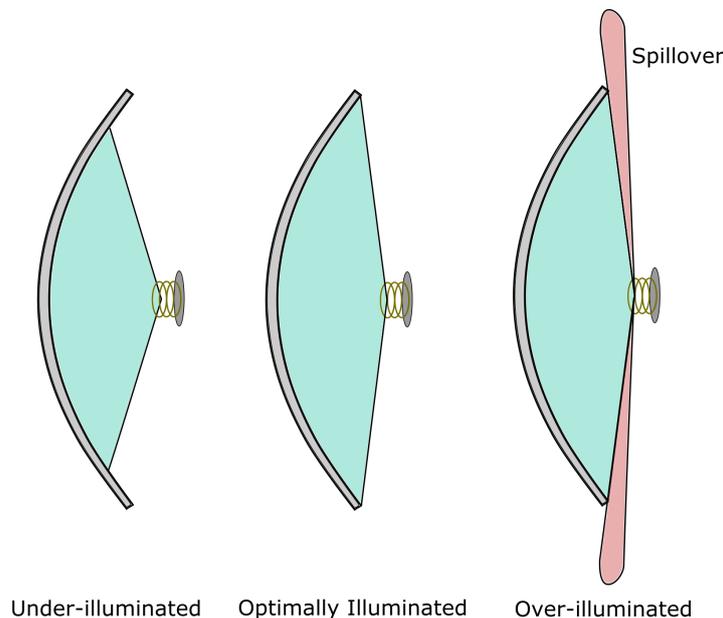


Figure 3.3: Optimizing reflector illumination

to be unused or underused, reducing efficiency and the effective surface area of the dish. In the case of over-illumination, the gain of the feed antenna is too low, i.e., its beamwidth is too wide, causing the beam to overshoot the dish and receive thermally radiated noise from the warm ground under the reflector, called “spillover”, making it harder to detect weak signals from space.

The third case is when the dish is optimally illuminated, which as a general rule is when the feed antenna pattern at the reflector’s rim is 10 dB less than it is at the center of the dish. [3, p.288] [4, p.385] At this point the efficiency losses from under-illumination are roughly balanced with the thermal noise generated by spillover. In some cases, such as for radio astronomy, it may be advantageous to intentionally under-illuminate the dish with up to a 15 dB edge taper because the reduction in effective reflector surface area is less critical than a reduction in thermal noise. [5]

As mentioned previously, a feed antenna must be designed to fit the dish it will be paired with and be optimized for its intended application. In order to get an idea of what kind of beamwidth the antenna will need, a bit of simple trigonometry can be performed based on dimensions pulled from the reflector’s datasheet, as shown in figure 3.4.

$$\alpha = \arctan\left(\frac{263 \text{ mm}}{1500 \text{ mm}}\right) \approx 10^\circ$$

3.1. THE PARABOLIC REFLECTOR

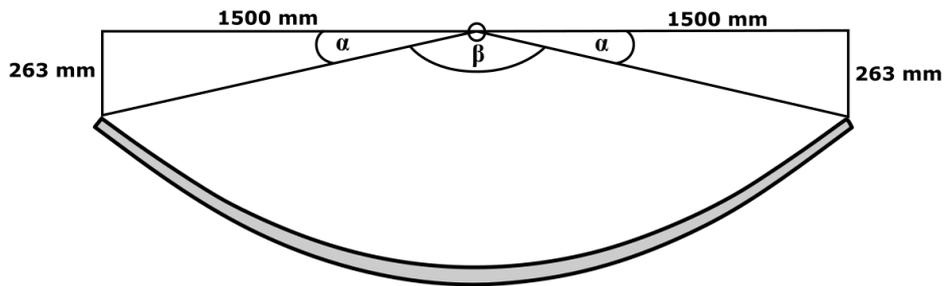


Figure 3.4: Calculating required antenna beamwidth

$$\beta = 180^\circ - 2 \cdot \alpha = 180^\circ - 20^\circ = 160^\circ$$

Thus, the approximate 10-15 dB beamwidth from rim-to-rim required is 160° , as measured by the feed antenna mounted at the focus. This will be an important factor in determining what type of antenna is feasible for use.

3.2 Antenna Selection

After an extensive literature research, three possible antenna types were selected then subsequently parametrically designed in CST Microwave Studio and simulated, a helical antenna, a horn antenna, and a circular patch antenna in air. While all three antennas would have done the job, the helical antenna was selected due to its small size, ease of construction, and robustness from minor tolerance differences.

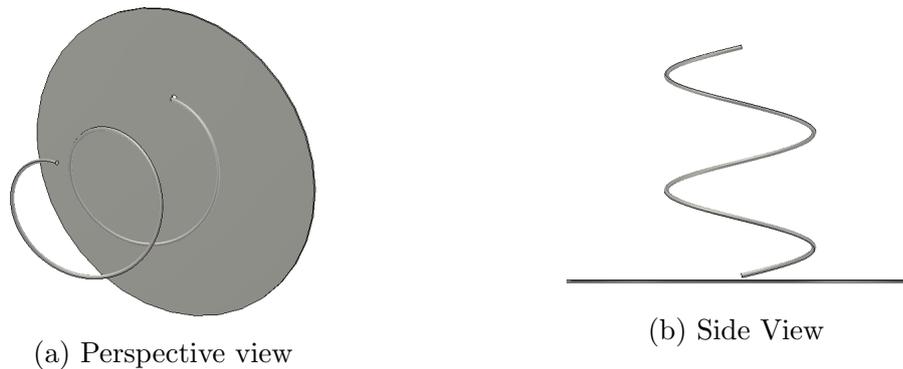


Figure 3.5: Helical antenna

A helical antenna consists primarily of two parts, a helix and a ground plane. The incoming electromagnetic wave is coupled onto the helix and travels down to the helical base where it enters a coaxial connector and is transported to the low-noise amplifier. In general, the gain of the antenna is dependent on the number of turns of the helix, the higher the number of turns, the higher the gain. The antenna will be mounted in the focus of the parabolic reflector where a motorized feed mount is located that can drive the antenna back and forth to focus the dish.



Figure 3.6: Motorized feed mount

3.3 Antenna Design

3.3.1 Helical Antenna Parameters

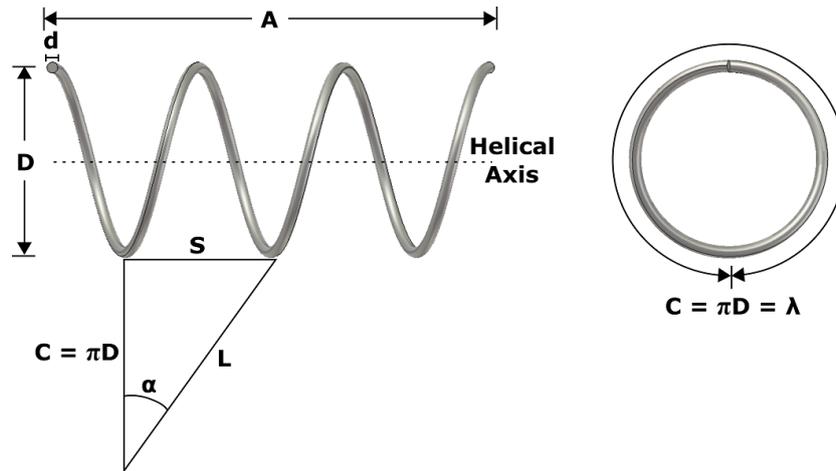


Figure 3.7: Helix dimensions

The Helix and its dimensions: [3, p.229]

D = diameter of helix (center to center)

C = circumference of helix = πD

S = spacing between turns (center to center)

α = pitch angle = $\arctan(\frac{S}{\pi D})$

L = length of 1 turn

n = number of turns

A = axial length = nS

d = diameter of helix conductor

The antenna will be designed around a center frequency of 1.42 GHz for the hydrogen line, therefore the wavelength is:

$$\lambda = \frac{c_0}{f} = \frac{300e6 \frac{m}{s}}{1.42 \text{ GHz}} = 211 \text{ mm}$$

For the antenna to operate in axial-fire mode, the circumference C of the helix must be between $0.8\lambda - 1.2\lambda$, centering this at 1.0λ for initial calculations gives us: [3, p.230]

$$C = 1.0\lambda = 1.0 \cdot 211 \text{ mm} = 211 \text{ mm}$$

3.3. ANTENNA DESIGN

$$D = \frac{C}{\pi} = \frac{211 \text{ mm}}{\pi} = 67 \text{ mm}$$

The pitch angle, α , for axial-fire mode is generally restricted to $12 - 14^\circ$, centering this at 13° for an initial calculation: [3, p.232]

$$\alpha = 13^\circ$$

$$S = \tan(\alpha) \cdot \pi D = \tan(13^\circ) \cdot \pi \cdot 67 \text{ mm} = 48.7 \text{ mm}$$

To calculate the number of turns, n , required, one must know the angular width the antenna sitting in the focus would measure from edge to edge of the parabolic reflector. As calculated in 3.1, this is approximately 160° . With this information in hand, a 10 dB edge-taper can be estimated with: [3, p.288]

$$n \simeq \frac{8400}{\phi^2 S_\lambda} \simeq \frac{8400}{(160^\circ)^2 \cdot 0.231} \simeq 1.4 \text{ turns}$$

where

$$\phi = 10 \text{ dB beamwidth in degrees} = 160^\circ$$

$$S_\lambda = \text{turn spacing in wavelengths} = \frac{S}{\lambda} = \frac{48.7 \text{ mm}}{211 \text{ mm}} = 0.231$$

With the number of turns known, the axial length, A , can be calculated:

$$A = nS = 1.4 \cdot 48.7 \text{ mm} = 68.2 \text{ mm}$$

Conductor diameter d is generally uncritical within the range of $0.005\lambda - 0.05\lambda$. [3, p.231]

$$d = 0.005\lambda - 0.05\lambda = 0.005 \cdot 211 \text{ mm} - 0.05 \cdot 211 \text{ mm} \approx 1 \text{ mm} - 10 \text{ mm}$$

Now that the helix is fully defined, we can turn our attention to the ground plane. In the past, helical antennas have been outfitted with a wide variety of different ground planes, ranging from basic to exotic, and in some cases none at all. Due to the circular symmetry of the parabolic reflector and the very wide beamwidth requirement, a simple circular ground plane was chosen. A good starting point according to Kraus [3, p.232] is a diameter D_{gp} of the ground plane to be $\frac{3}{4}\lambda$.

$$D_{gp} = \frac{3}{4}\lambda = \frac{3}{4} \cdot 211 \text{ mm} = 158 \text{ mm}$$

3.3.2 Modeling the Antenna

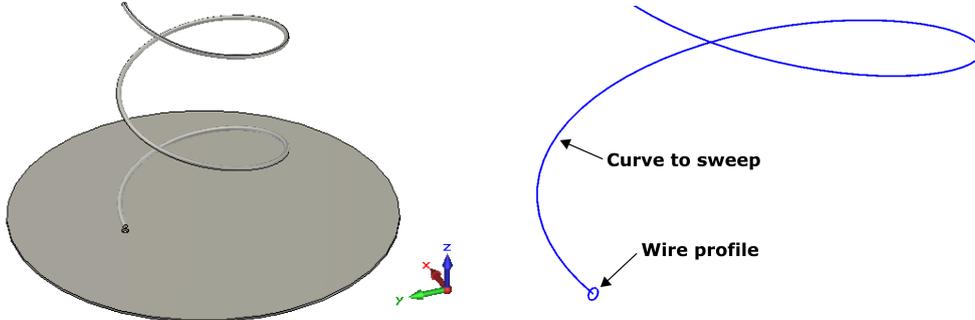


Figure 3.8: Basic modeling of a helical antenna

After running through all of the necessary design equations to get a first estimation of the antenna's parameters, we can now move to parametrically modeling the antenna in an electromagnetic simulation software package, in this case, CST Microwave Studio.

Parametric modeling is describing the dimensions of a 3D model using descriptive variables so that major changes can be automatically generated in the model just by changing one of the variables. This allows for testing of variable dependencies and later an automated optimization process.

Using the variables D , n and A as described in section 3.3.1, a 2 dimensional curve in 3D space can be modeled to represent the helix with:

$$\begin{aligned}x(t) &= \frac{D}{2} \cdot \sin(n \cdot 2\pi \cdot t) \\y(t) &= \frac{D}{2} \cdot \cos(n \cdot 2\pi \cdot t) \\z(t) &= offset_{gp} + t \cdot A \\t &\in [0, 1]\end{aligned}$$

as the helix is simply a circle with diameter D continuously rotating in space up to height $A + offset_{gp}$. The variable t is simply the independent parametric variable that is swept from 0 to 1 to define the curve.

Once the analytical curve is defined, a circular outline of radius $\frac{d}{2}$ can be swept along the 2D curve to give the helix volume, making it a wire.

The model for the ground plane is much simpler, as it is a basic cylinder with a height of H_{gp} , a radius of $\frac{D_{gp}}{2}$ and a circular hole cut for the feed at helical radius $\frac{D}{2}$ on the y -axis.

3.3.3 Modeling the Parabolic Reflector

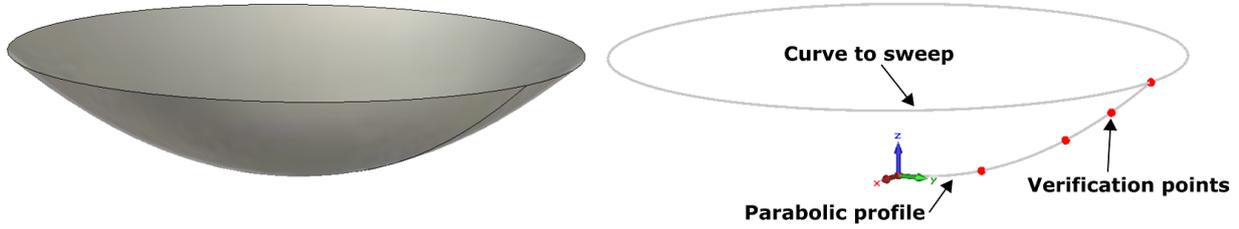


Figure 3.9: Modeling the parabolic reflector

Mathematically, a parabolic reflector can be fully defined by two parameters, the diameter D and the focal point f . The equation defining the parabola can be described as: [8, p.30]

$$z(y) = \frac{y^2}{4f} \quad (3.1)$$

Using this equation, a parabolic profile can be modeled in CST Microwave Studio and be swept along a circular curve that defines the rim of the reflector. In the coordinate system shown in figure 3.9 the parabolic profile can be analytically expressed as:

$$\begin{aligned} y(t) &= \frac{D}{2} \cdot t \\ z(y) &= \frac{y^2}{4f} \\ t &\in [0, 1] \end{aligned}$$

The rim of the reflector, being the curve over which the profile will be swept, can be defined as:

$$\begin{aligned} x(t) &= \frac{D}{2} \cdot \cos(t) \\ y(t) &= \frac{D}{2} \cdot \sin(t) \\ z &= \frac{(\frac{D}{2})^2}{4f} \\ t &\in [0, 2\pi] \end{aligned}$$

To verify that this model does actually fit the real parabolic reflector, five verification points were taken from the reflector's datasheet and mapped along the curve.

3.3.4 Impedance Matching

Helical antennas have a characteristic impedance of approximately 120-140 Ω and are nearly purely resistive. In order to use such an antenna efficiently in a standard 50 Ω system the impedance needs to be transformed to a value much closer to 50 Ω .

There are several techniques for impedance matching, but one of the most common and practical is the Quarter-Wave ($\lambda/4$) transformer.

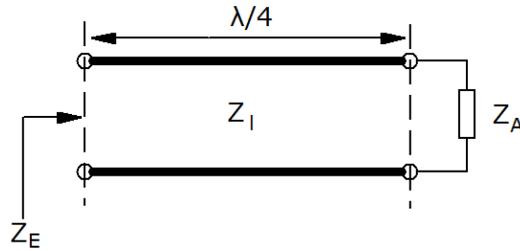


Figure 3.10: Quarter-Wave transformer concept

$$Z_l = \sqrt{Z_A \cdot Z_E} \quad [2, p.160] \quad (3.2)$$

Put simply, if we have a 130 Ω antenna (Z_A) and need to match it to a 50 Ω system (Z_E), then the impedance of the quarter-wave transformer is:

$$Z_l = \sqrt{Z_A \cdot Z_E} = \sqrt{130\Omega \cdot 50\Omega} = 80.6 \Omega$$

In order to achieve this desired impedance, a brass microstrip line in air was conceived running from the base of the helix to the 50 Ω SMA connector. Using Keysight ADS's LineCalc software with an $\epsilon_r = 1$ for air, a width of 10 mm, and a thickness of 1 mm, it can be calculated that a height variation of 3-7 mm could produce a range of impedances from 60-100 Ω . Using a cutout in the antenna's support, small 1 mm thick 3D printed blocks could be stacked to vary the height so that the antenna could be matched experimentally with a vector network analyzer. Coincidentally, the circumference of the helix is 1 λ , making the $\lambda/4$ strip 1/4th of the helix's circumference.

The match of the antenna was measured with a vector network analyzer, VNA, showing an 800 MHz bandwidth with a reflection factor less than -10 dB. At our frequency of interest, 1420 MHz, it shows an s_{11} match of -13 dB.

3.3. ANTENNA DESIGN

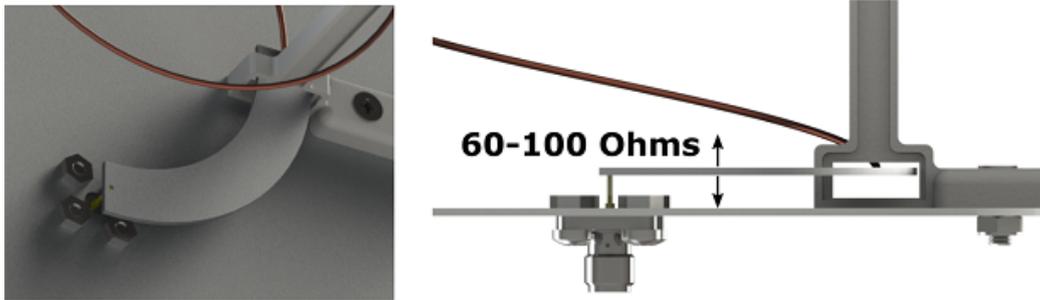


Figure 3.11: Quarter-wave transformer



Figure 3.12: Reflection factor s_{11} as measured by VNA

3.3.5 Mechanical Design

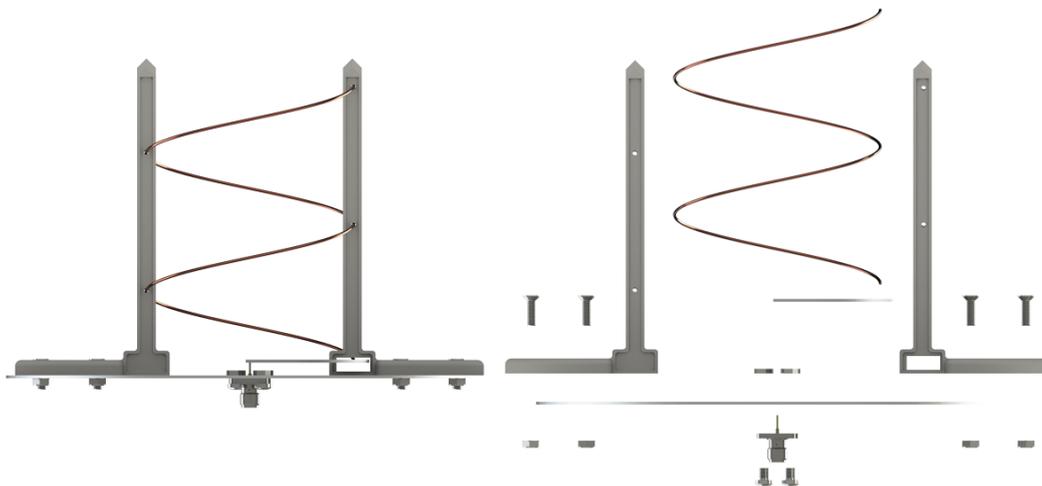


Figure 3.13: Rendered design

After the helical antenna was designed, optimized, and a $\lambda/4$ strip was added in CST Microwave Studio, it was exported as a .STEP file and imported into a mechanical CAD program. Supports were designed for the helix and transformer that would be 3D printed from ABS plastic. ABS plastic has a poorly defined dielectric constant ϵ_r and thus transitions through it should be kept to a minimum. To achieve this, cutouts in the frame of the plastic allowed the area where the helix passed through to be as small as possible (2 mm) while still maintaining mechanical strength (4 mm thick frame). Kraus suggests dielectric supports can be a few hundredths of a wavelength without noticeable effect. [3, p.231] ($\frac{2 \text{ mm}}{211 \text{ mm}} = 0.009$)

Once the mechanical design was completed, the antenna was exported again in a .STEP file and re-imported into CST Studio. The biggest value to using .STEP files over other types of files is that every object's individuality is preserved, so that individual material properties can be assigned. The new supports were assigned ABS plastic, the ground plane aluminum, and the $\lambda/4$ strip as brass.

During the 3D printing process it became obvious that the mechanical stability of the original helix supports was inadequate as they would snap like twigs. To fight this, the supports were adjusted and reprinted, making their frame larger and thicker, while only increasing the center inset thickness where the coil passes through by an extra millimeter (bringing it to 3 mm). This showed a dramatic increase in mechanical strength and had no serious effect on the antenna's performance.

3.4 Antenna Simulation

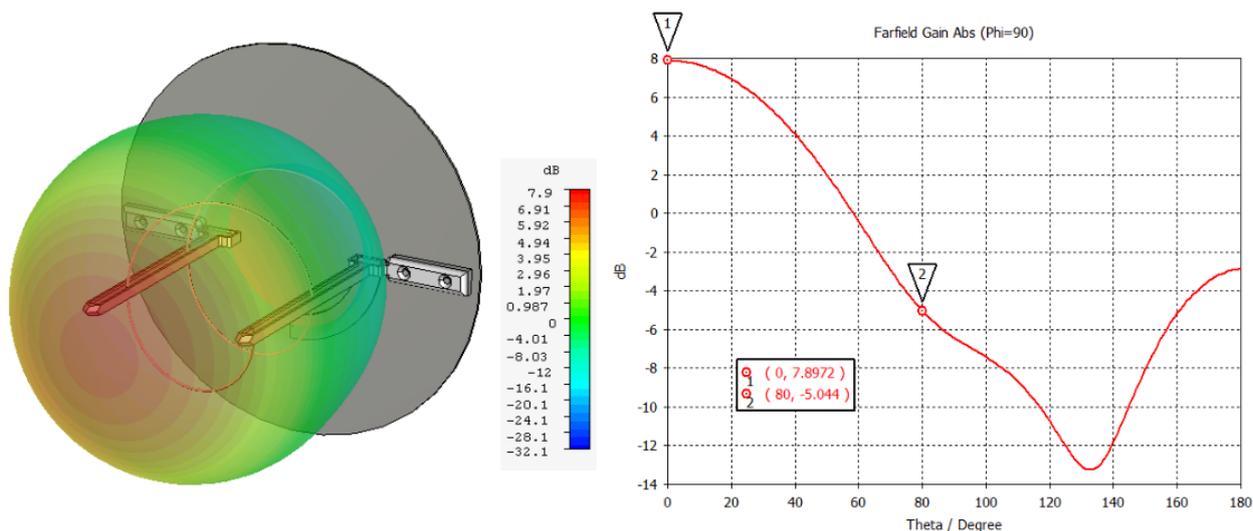


Figure 3.14: Simulation of completed antenna

The antenna was simulated in CST Microwave Studio using the Integral Solver. The antenna's main-lobe is along the helical axis and has about 8 dB of gain. At an angular distance of 80° , which corresponds to the rim of the parabolic reflector, it shows a gain of -5 dB. This means that the edge taper of the antenna / reflector combination is $8 \text{ dB} - (-5 \text{ dB}) = 13 \text{ dB}$, meaning a slight under-illumination to minimize picking up radiation from the warm ground under the radio telescope.

At this point one of the major advantages of a helical antenna becomes obvious. There is only one side-lobe and it is at 180° from the main-lobe, meaning it is pointed towards space and not toward the ground or a nearby cell tower.

The originally calculated number of turns, 1.4, showed a poor front-to-back ratio. A helical antenna first begins to act like a helical antenna around 1.8 turns. A value of 2 turns showed to be optimal for this design.

3.4.1 Antenna and Reflector Simulation

The antenna and parabolic reflector can now be simulated together to see their combined interaction. As can be seen in figure 3.15, the dish has a magnifying effect on the antenna, causing the combined gain of the system to jump to about 30 dB with a half-power beam-width, or HPBW of 5.6° .

3.4. ANTENNA SIMULATION

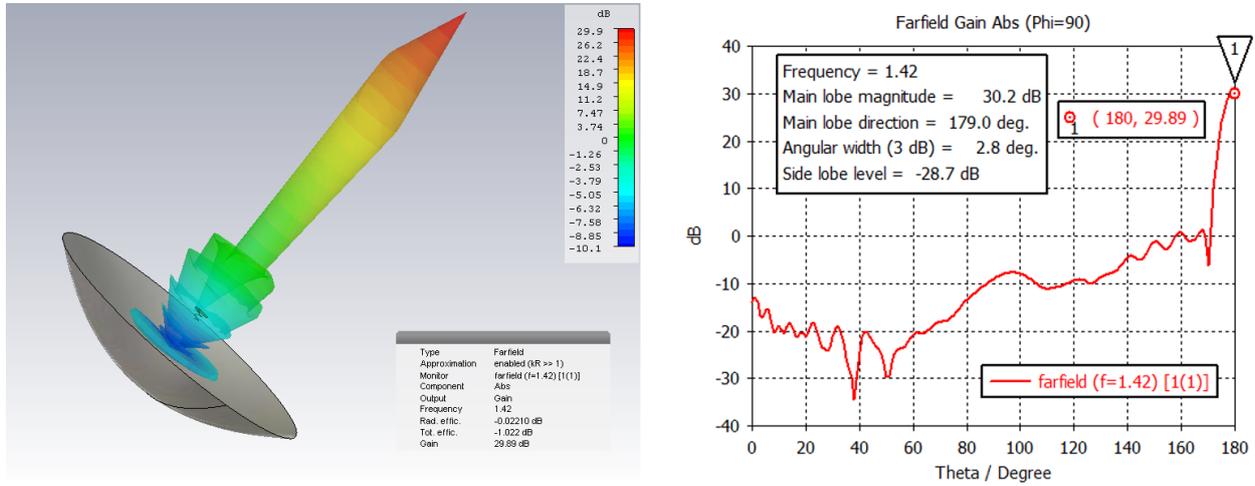


Figure 3.15: Simulation results of antenna in reflector's focus

The maximum, ideally achievable value for gain of a parabolic reflector is: [3, p.65]

$$G = \frac{4\pi A}{\lambda^2} = \frac{4\pi \frac{\pi d^2}{4}}{\lambda^2} = \frac{4\pi \frac{\pi 3m^2}{4}}{211 \text{ mm}^2} = 1,995 = 33 \text{ dB}$$

where A is the reflector's area and λ is the wavelength.

For the minimum, ideally achievable HPBW, the Rayleigh Criteria for circular apertures: [9, p.398]

$$\theta = 1.22 \frac{\lambda}{d} = 1.22 \frac{211 \text{ mm}}{3 \text{ m}} = 85.8 \text{ milliradian} = 4.9^\circ$$

Thus, the simulated values of the antenna/reflector combination are very close to the maximum achievable values. The gain is a little lower, and correspondingly the HPBW is a little broader.

One of the most notable results is the side-lobe level, SLL, of 30 dB, which is the ratio of gain of the main antenna lobe to the gain of the first side lobes. This means the signals picked up by the antenna from other directions are at least 1,000 times weaker than from the desired direction. The sharp point of the farfield tip comes from the resolution reduction to ease the immense computational time for the simulation.

3.5 Measurement Results

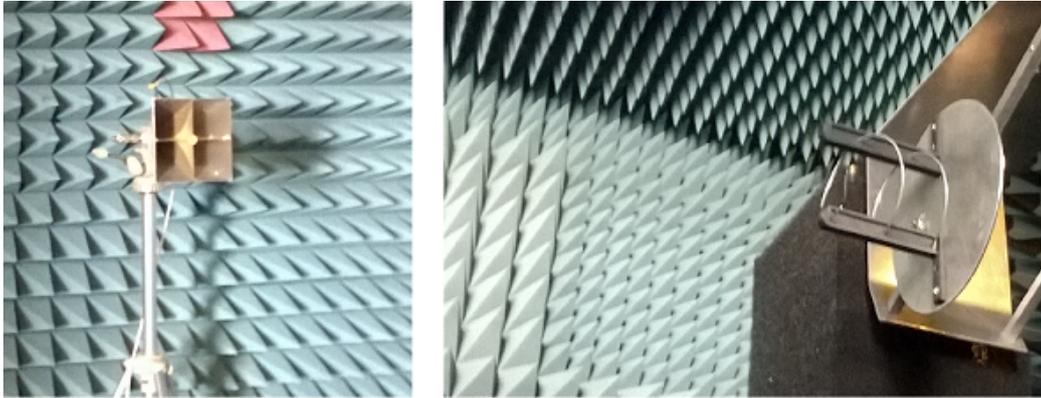


Figure 3.16: Testing in anechoic chamber

The antenna was measured in the Institute for Microwaves and Photonics' anechoic chamber, which is an extremely well shielded room with absorptive padding to reduce reflections. A very wide band vivaldi horn hooked up to a signal generator was used as the source. The antenna was mounted on an automated turntable and fed to a high-end spectrum analyzer. The measurement process was controlled by a LabView script. The antenna was successively rotated in 1 degree increments, measuring the received signal at each point. The actual farfield pattern closely resembles the simulated one.

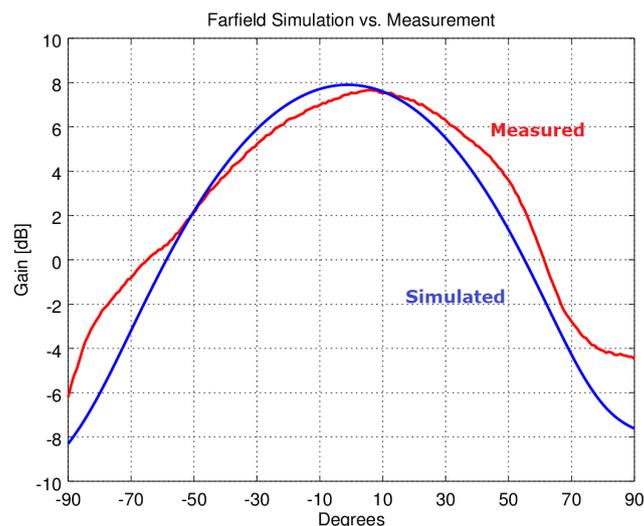


Figure 3.17: Measured farfield results vs. simulation

3.5. MEASUREMENT RESULTS



Figure 3.18: The finished antenna mounted on the reflector

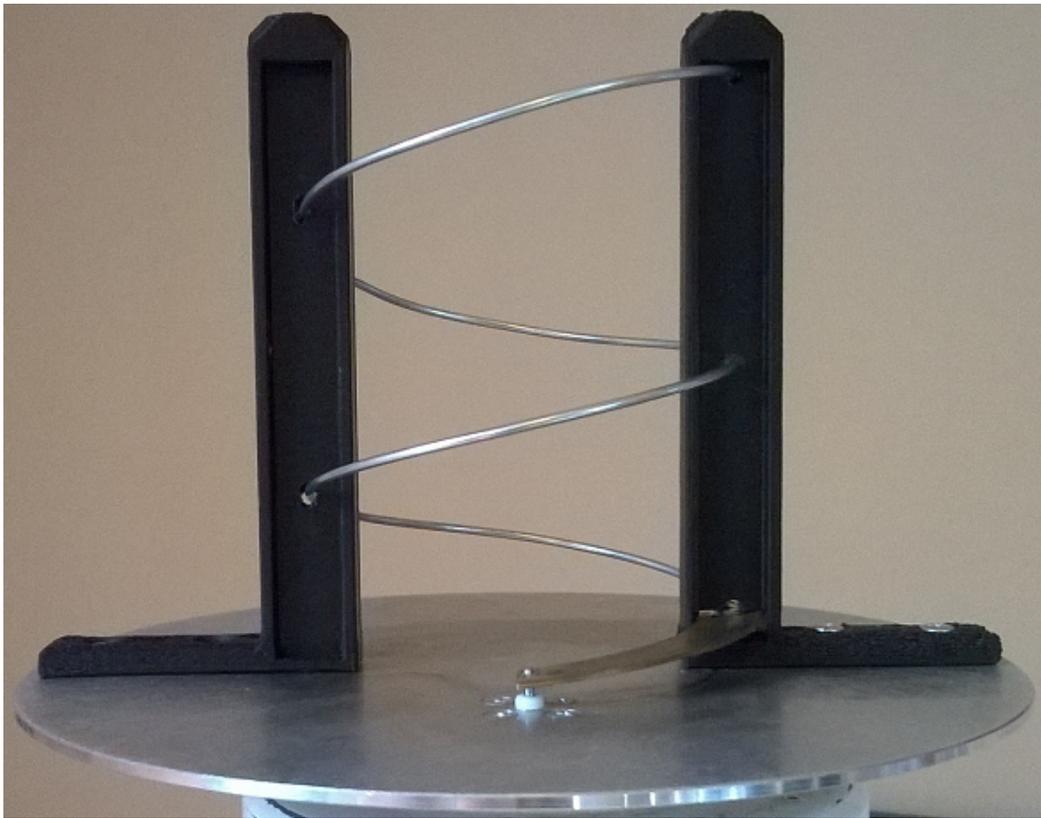


Figure 3.19: The designed antenna

Chapter 4

Preselect Filter

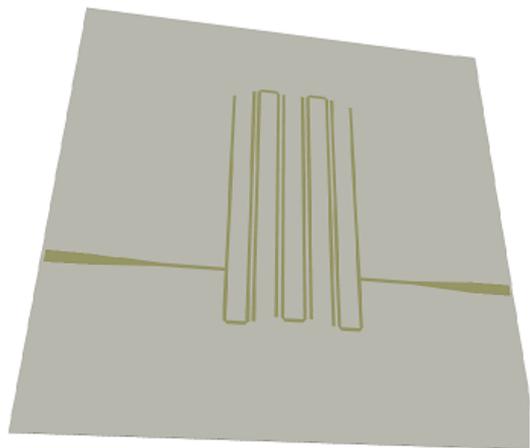
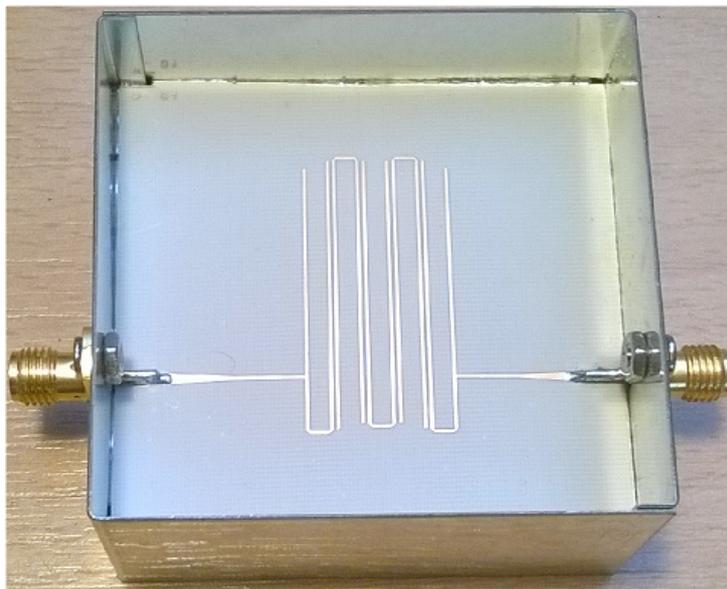


Figure 4.1: Hairpin microstrip bandpass filter

4.1 Overview

Any wireless receiver that needs to perform a down-conversion will need to have a band preselect filter placed before the first mixer in order to suppress the desired band's image frequencies. In this system, there are two possible intermediate frequencies, 150 MHz or 72.5 MHz, based on which footprint-compatible SAW filter is used in the analog front-end described in chapter 5.

In low-side injection for down-conversion, the image frequencies of the desired band appear at the local oscillator frequency minus the intermediate frequency, or $f_{LO} - f_{IF}$. This is shown in figure 4.2.

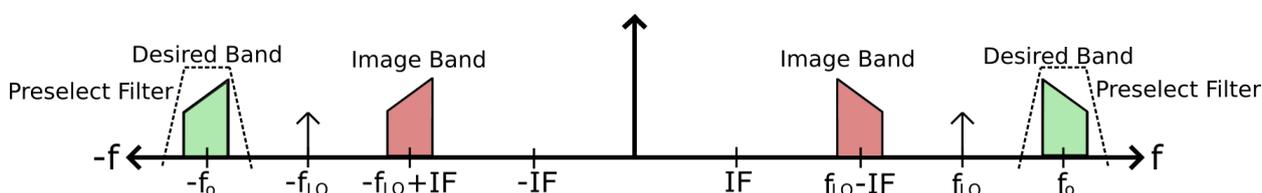


Figure 4.2: Spectrum pre-down-conversion

These image bands will always be mixed into our desired band and if they aren't adequately attenuated it will destroy our signal and it can not be recovered. This is particularly important for a radio telescope as the transmission from a satellite flying overhead will be dramatically more powerful than the incredibly weak signal we're trying to receive from across the galaxy.

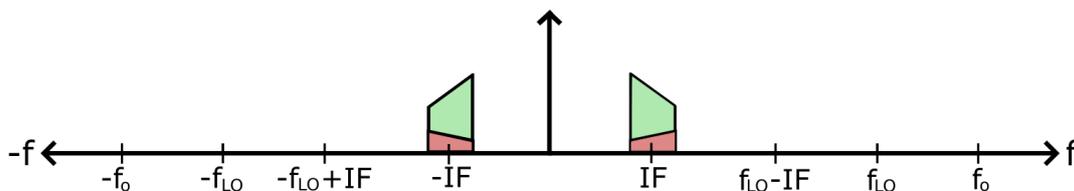


Figure 4.3: Spectrum post-down-conversion with preselect filter

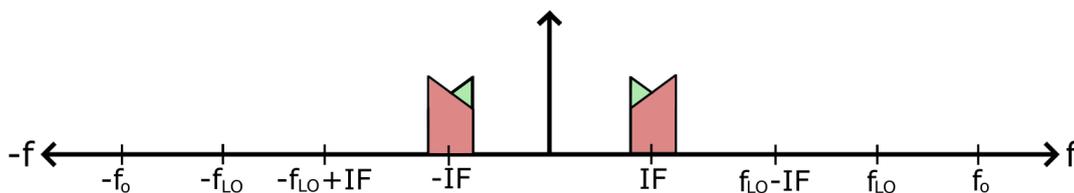


Figure 4.4: Spectrum post-down-conversion without preselect filter

4.2 Filter Synthesis

The typical filter design process is lengthy and is a science unto itself. The process used here is shown in figure 4.5.

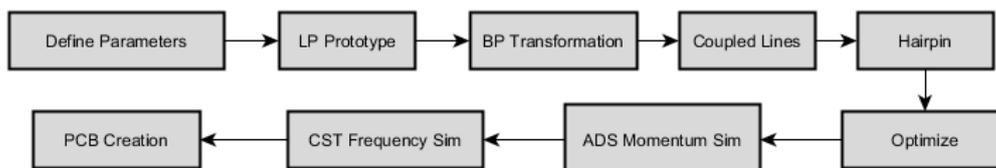


Figure 4.5: The filter synthesis design process

4.2.1 Define Parameters

Initially, a center frequency of 1.42 GHz was chosen, however, after experimentation, it became evident that intentionally off-setting the filter to a slightly higher frequency could provide a higher rejection of the image bands at the lower frequencies as mentioned in section 4.1.

The fractional bandwidth is the ratio of the bandwidth of the filter to its center frequency. This is an important parameter as the bandwidth alone can be misleading. In general in filter design, the higher the center frequency, the harder it is to make a very narrow filter. The battle being waged is always between insertion loss and bandwidth. The narrower the filter, i.e., the lower fractional bandwidth, the higher the insertion loss will be, and vice versa. For this reason one can not pick arbitrarily low values for fractional bandwidth. This process was carried out for fractional bandwidths of 10%, 15%, and 20%, and it was found that 15% provided the best balance between insertion loss and bandwidth.

Center Frequency: $f_0 = 1.47$ GHz

Fractional Bandwidth: $\delta = \frac{\Delta f}{f_0} = 15\%$

Reference Impedance: 50Ω

4.2.2 Low-Pass Prototype

A common first step in developing a bandpass filter is to first start with an abstract, discrete low-pass filter and then perform a band-pass transformation. For this, a standard canonical LC network is chosen.

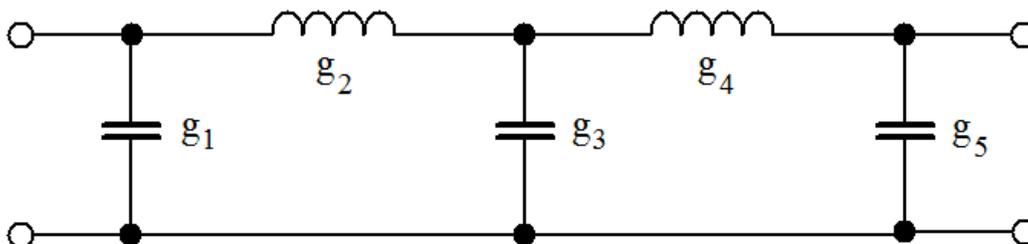


Figure 4.6: Canonical LC network, Cauer Topology, 5th order

Individual component values can be calculated with: [1, p.106]

$$L_i = \frac{g_i Z_n}{2\pi f_c} \quad (4.1)$$

$$C_i = \frac{g_i}{Z_n 2\pi f_c} \quad (4.2)$$

The g_i values are obtained from a reference table, whose parameters are based on the filter type (e.g. Butterworth, Chebyshev, etc.), desired pass-band ripple, and filter order. For this filter a 5th order Chebyshev Filter with a pass-band ripple of $A_c = 0.5$ dB was chosen.

Table 4.1: Chebyshev TP $A_c = 0.5$ dB [1, p.108]

n	g_1	g_2	g_3	g_4	g_5	g_{n+1}
1	0.6986					1
2	1.4029	0.7071				1.9841
3	1.5963	1.0967	1.5963			1
4	1.7603	1.1926	2.3661	0.8419		1.9841
5	1.7058	1.2296	2.5408	1.2296	1.7058	1

Using these values and equations 4.1 - 4.2, we can calculate the capacitances and inductances for the low-pass prototype:

$$g_{1,5} = 1.7058 \Rightarrow C_{1,5} = \frac{g_i}{Z_n 2\pi f_c} = \frac{1.7058}{50\Omega \cdot 2\pi \cdot 110\text{MHz}} = 49.4 \text{ pF}$$

$$g_3 = 2.5408 \Rightarrow C_3 = \frac{g_i}{Z_n 2\pi f_c} = \frac{2.5408}{50\Omega \cdot 2\pi \cdot 110\text{MHz}} = 73.5 \text{ pF}$$

4.2. FILTER SYNTHESIS

$$g_{2,4} = 1.2296 \Rightarrow L_{2,4} = \frac{g_i Z_0}{2\pi f_c} = \frac{1.2296 \cdot 50\Omega}{2\pi \cdot 110\text{MHz}} = 89.0 \text{ nH}$$

The filter is then built and simulated in Keysight ADS:

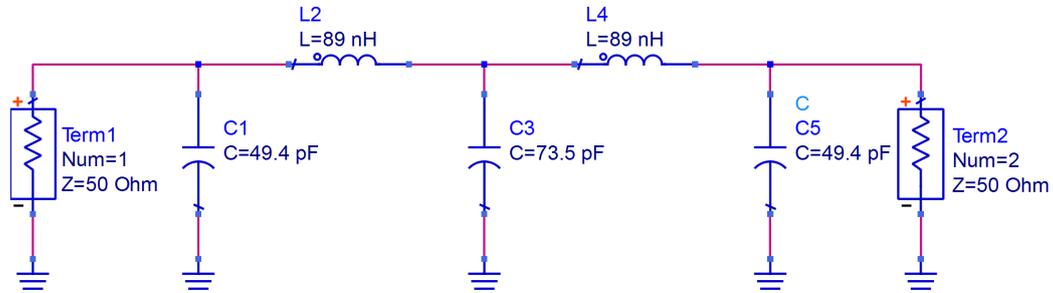


Figure 4.7: Synthesized low-pass LC filter

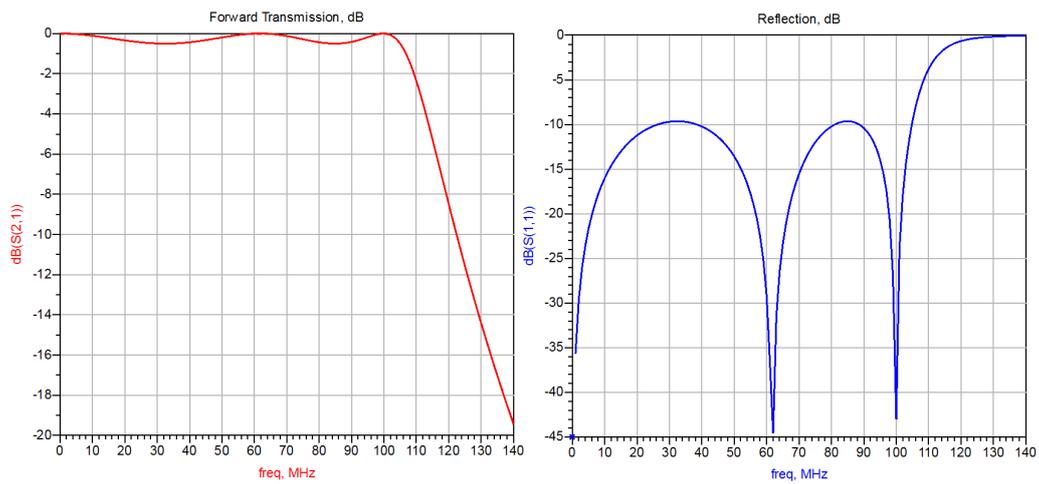


Figure 4.8: Simulated results

4.2.3 Band-Pass Transformation

The next step in the filter synthesis process is to perform a Low-Pass \Rightarrow Band-Pass transformation.

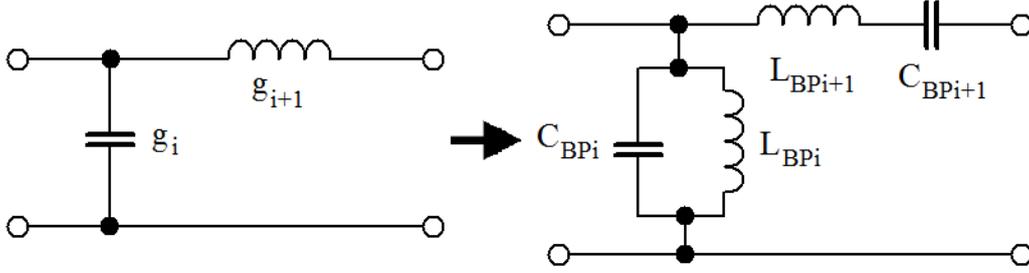


Figure 4.9: Band-pass transformation

The required values for each stage can be calculated with: [1, p.110]

$$C_{BP_i} = \frac{g_i}{2\pi\Delta f \cdot Z_n} \quad L_{BP_i} = \frac{\delta \cdot Z_n}{2\pi f_0 \cdot g_i} \quad (4.3)$$

$$C_{BP_{i+1}} = \frac{\delta}{g_{i+1} \cdot Z_n \cdot 2\pi f_0} \quad L_{BP_{i+1}} = \frac{g_{i+1} \cdot Z_n}{2\pi\Delta f} \quad (4.4)$$

Inserting the applicable parameters the transformed values can be calculated:

$$C_1 = C_5 = \frac{g_1}{2\pi\Delta f \cdot Z_n} = \frac{1.7058}{2\pi \cdot 220\text{MHz} \cdot 50\Omega} = 24.7 \text{ pF}$$

$$L_1 = L_5 = \frac{\delta \cdot Z_n}{2\pi f_0 \cdot g_1} = \frac{0.15 \cdot 50\Omega}{2\pi \cdot 1.47\text{GHz} \cdot 1.7058} = 476 \text{ pH}$$

$$C_2 = C_4 = \frac{\delta}{g_2 \cdot Z_n \cdot 2\pi f_0} = \frac{0.15}{1.2296 \cdot 50\Omega \cdot 2\pi \cdot 1.47\text{GHz}} = 264 \text{ fF}$$

$$L_2 = L_4 = \frac{g_2 \cdot Z_n}{2\pi\Delta f} = \frac{1.2296 \cdot 50\Omega}{2\pi \cdot 220\text{MHz}} = 44.5 \text{ nH}$$

$$C_3 = \frac{g_3}{2\pi\Delta f \cdot Z_n} = \frac{2.5408}{2\pi \cdot 220\text{MHz} \cdot 50\Omega} = 36.8 \text{ pF}$$

$$L_3 = \frac{\delta \cdot Z_n}{2\pi f_0 \cdot g_3} = \frac{0.15 \cdot 50\Omega}{2\pi \cdot 1.47\text{GHz} \cdot 2.5408} = 320 \text{ pH}$$

Building and simulating this filter in Keysight's ADS:

4.2. FILTER SYNTHESIS

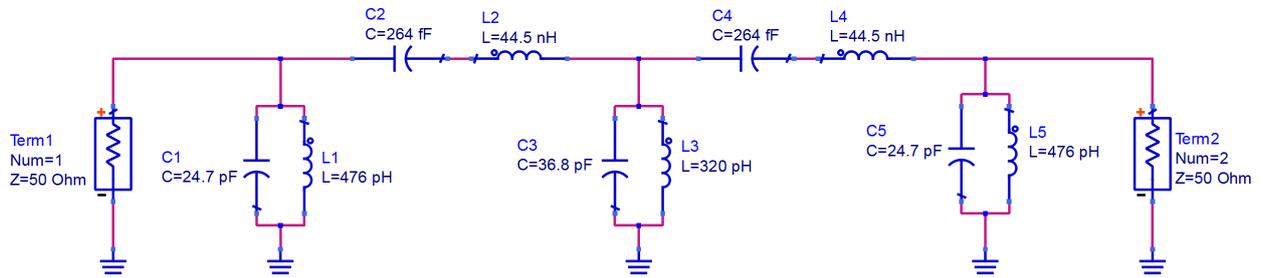


Figure 4.10: Synthesized band-pass LC filter

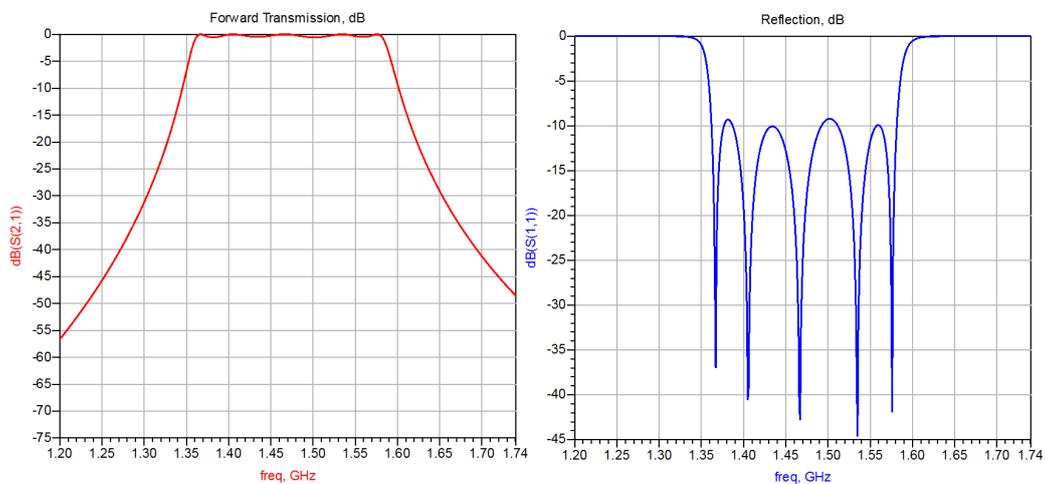


Figure 4.11: Simulated results

4.2.4 Coupled-Line Transformation

Now that the band-pass filter has been generated as an ideal LC network, it must be converted into a real, coupled-line microstrip implementation. Each LC combination from the previous filter represents one pair of coupled-lines. The parameters are line width, length and spacing, as shown in figure 4.12:

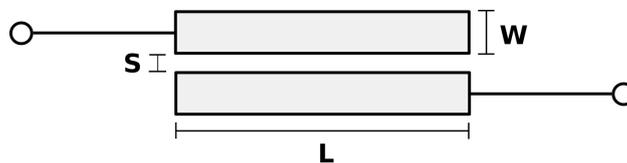


Figure 4.12: Coupled-line microstrip parameters

To determine these parameters one must first calculate the even- and odd-mode line impedances for each LC combination. This can be approximated

4.2. FILTER SYNTHESIS

with the following equations: [10, p.431]

$$Z_{e_n} = Z_o[1 + J_n Z_o + (J_n Z_o)^2] \quad (4.5)$$

$$Z_{o_n} = Z_o[1 - J_n Z_o + (J_n Z_o)^2] \quad (4.6)$$

Where each $J_n Z_o$ combination can be approximated with: [10, p.431]

$$J_1 Z_o \approx \sqrt{\frac{\pi \delta}{2g_1}} \quad (4.7)$$

$$J_n Z_o \approx \frac{\pi \delta}{2\sqrt{g_n \cdot g_{n-1}}} \quad (4.8)$$

$$J_{N+1} Z_o \approx \sqrt{\frac{\pi \delta}{2g_N \cdot g_{N+1}}} \quad (4.9)$$

Using equations 4.5 - 4.9 a table of the even- and odd-impedances can be calculated.

For example with $n = 2$:

$$J_2 Z_o \approx \frac{\pi \delta}{2\sqrt{g_2 \cdot g_1}} = \frac{\pi \cdot 0.15}{2\sqrt{1.2296 \cdot 1.7058}} = 0.163$$

$$Z_{e_2} = Z_o[1 + J_2 Z_o + (J_2 Z_o)^2] = 50\Omega[1 + 0.163 + (0.163)^2] = 59.5 \Omega$$

$$Z_{o_2} = Z_o[1 - J_2 Z_o + (J_2 Z_o)^2] = 50\Omega[1 - 0.163 + (0.163)^2] = 43.2 \Omega$$

Table 4.2: Even- and Odd-impedances

n	g_n	$J_n Z_o$	Z_{e_n}	Z_{o_n}
1	1.7058	0.372	75.5	38.3
2	1.2296	0.163	59.5	43.2
3	2.5408	0.133	57.5	44.2
4	1.2296	0.133	57.5	44.2
5	1.7058	0.163	59.5	43.2
6	1.7058	0.372	75.5	38.3

It is at this point that one must consider what type of substrate will be used and what its parameters are. Most low-frequency circuit boards are manufactured in FR4, however due to its high loss-tangent ($\tan(\delta) \geq 0.02$),

4.2. FILTER SYNTHESIS

the losses at microwave frequencies can become unacceptable, especially if a filter with a high Q-factor is desired.

An attractive alternative to FR4 is Rogers 4003c™ substrate which is manufactured from a proprietary woven glass reinforced hydrocarbon material. The loss-tangent is about 1/10th that of FR4 ($\tan(\delta) \approx 0.0027$), thus making it much more efficient but also dramatically more expensive.

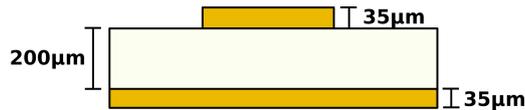


Figure 4.13: 2-layer Rogers 4003c™ substrate

Once all of the odd- and even-impedances have been calculated and a suitable substrate has been selected, a microstrip calculator such as Keysight's LineCalc can be used to calculate W, S, and L parameters of each of the coupled lines.

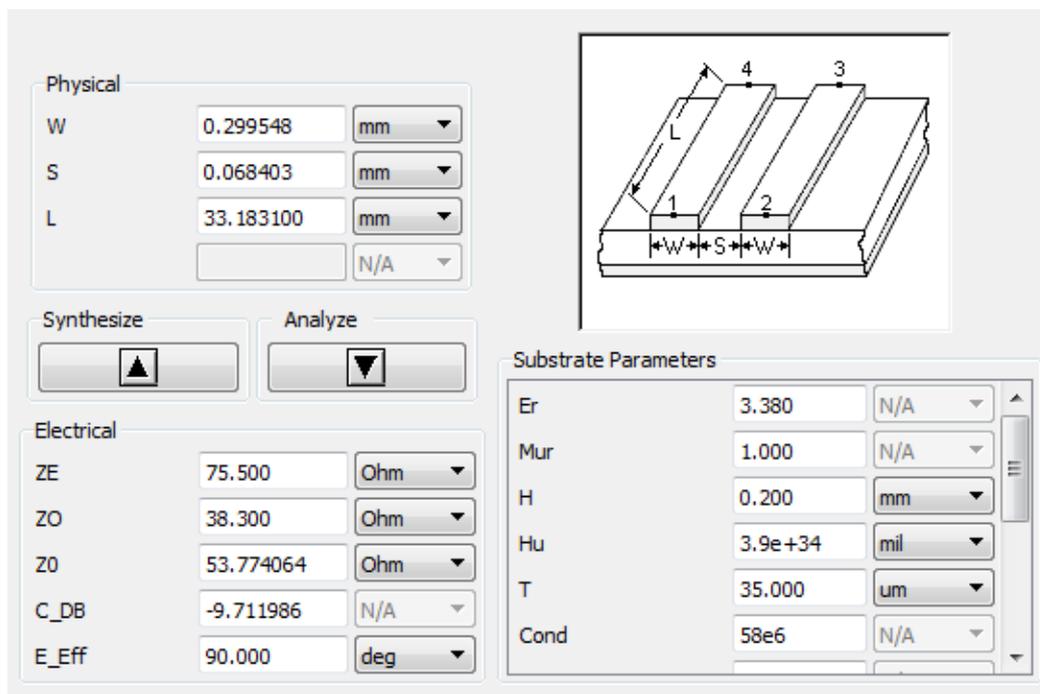


Figure 4.14: Screenshot of LineCalc Software from Keysight

4.2. FILTER SYNTHESIS

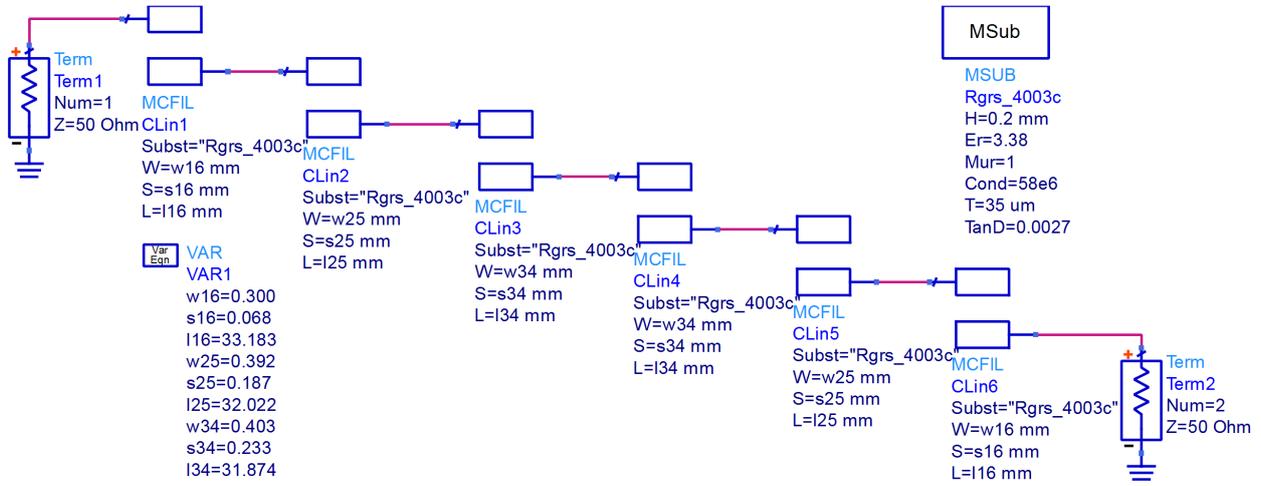


Figure 4.15: Coupled-line microstrip schematic in Keysight ADS



Figure 4.16: Coupled-line microstrip band-pass filter

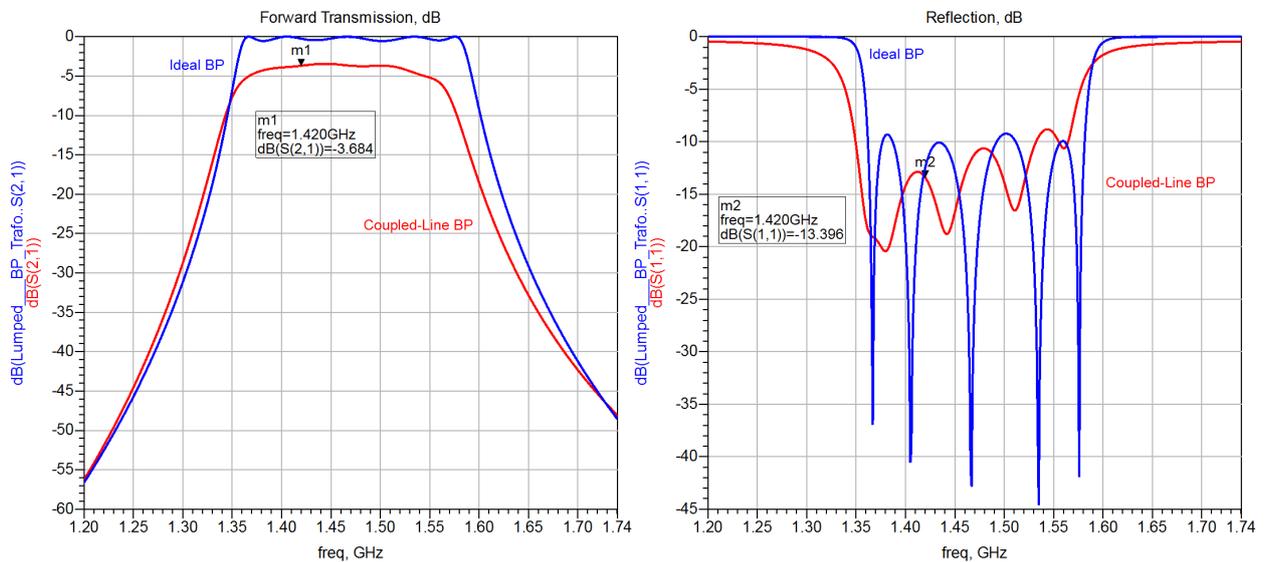


Figure 4.17: Ideal / real data comparison

4.2.5 Hairpin Filter

Coupled-line microstrip filters are simple and effective, but without some modification they tend to be very long. The filter designed in 4.2.4 was about 22 cm long, an absurdly impractical dimension that would send the cost and size of the board through the roof. In order to overcome this, the structure is bent into a U-shape, or as the name implies, the shape of a hairpin.

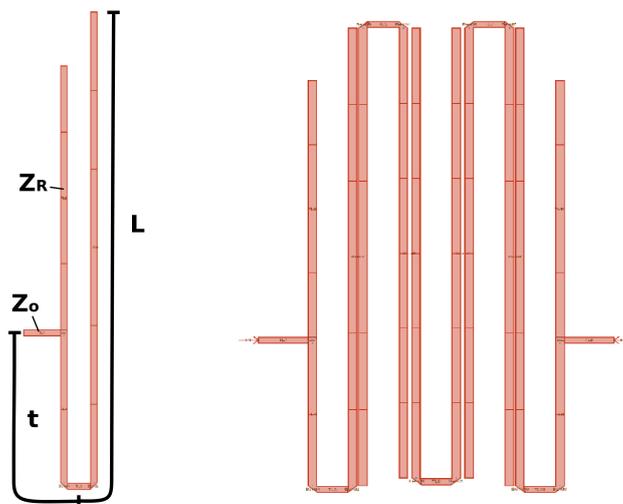


Figure 4.18: Hairpin filter dimensions and layout

The L dimension in figure 4.18 is the length of the individual coupled-line pairs from 4.2.4. In this form, the length of the actual coupled-line element in ADS needs to be slightly reduced to compensate for the extra length at the bottom of the U.

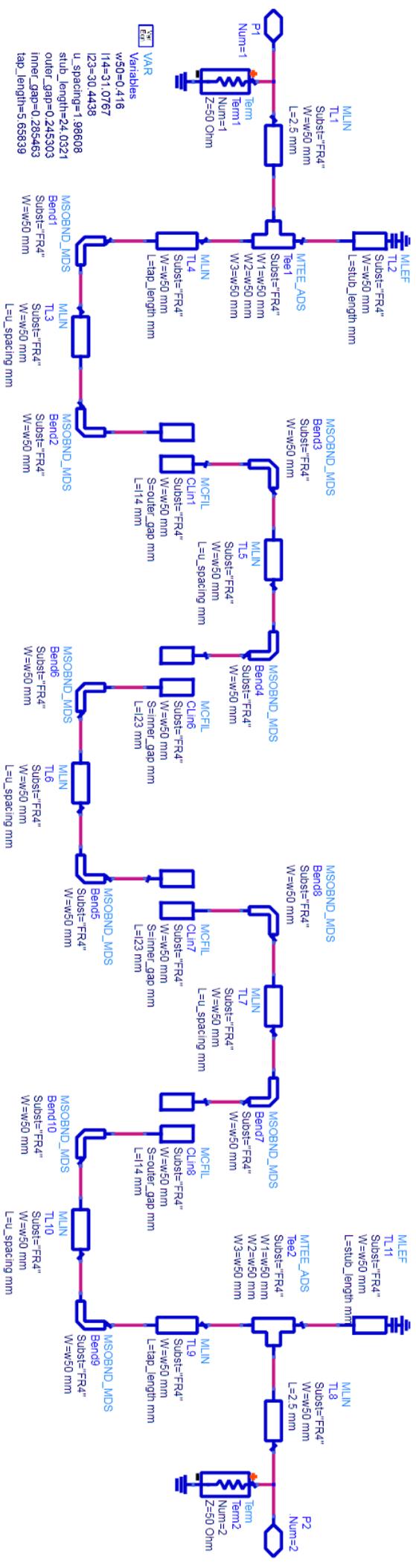
The tap-length t as pictured in figure 4.18 is the primary characteristic for determining the input and output impedances of the filter. For this case, we want 50Ω , which can be approximated with: [11, p.131]

$$t \approx \frac{2L}{\pi} \cdot \sin^{-1} \left(\sqrt{\frac{\pi(Z_o/Z_R)}{2Q_e}} \right) \approx \frac{2 \cdot 28.7 \text{ mm}}{\pi} \cdot \sin^{-1} \left(\sqrt{\frac{\pi(50\Omega/50\Omega)}{2 \cdot 11.37}} \right) \approx 6.96 \text{ mm}$$

The Q-Factor can be approximated with: [11, p.129]

$$Q_e \approx \frac{g_0 \cdot g_1}{\delta} \approx \frac{1 \cdot 1.7058}{0.15} \approx 11.37$$

The schematic and its values can be seen on the next page.



4.3 Simulation and Optimization

4.3.1 Optimization

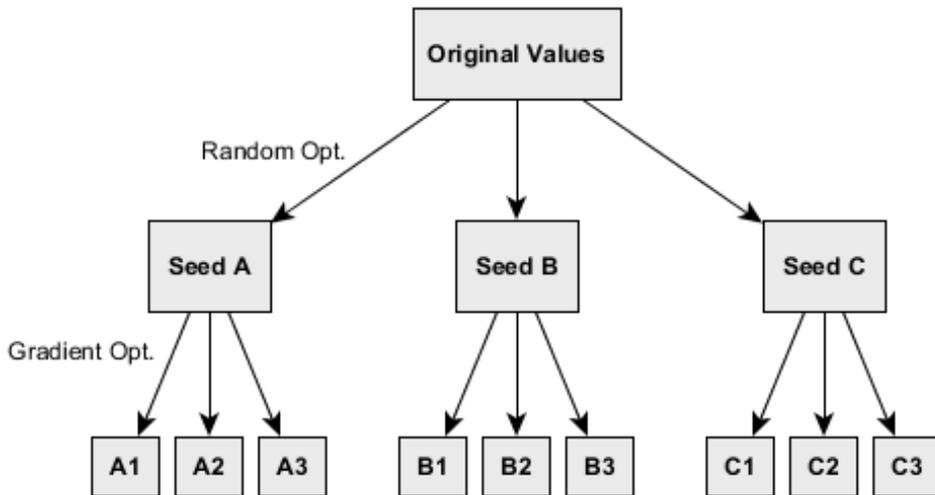


Figure 4.19: The optimization process

Once the schematic design is parameterized and simulation shows that the values are reasonably close, it is time to begin a numeric optimization process. The use of numeric optimization has risen dramatically in the past several decades due to ever-increasing computational capacity of modern computers and the rising complexity of designs. In most cases in microstrip design, analytically solving the field equations to get meaningful data is either incredibly difficult and time consuming or outright impossible.

Numeric optimization is not without its challenges. Unfortunately one can not simply blindly set goals and press “GO” to make any real progress. As such, it is important to have a deep understanding of how the optimization algorithms function, what their strengths and weaknesses are, and to apply them in an intelligent and logical process.

In the design of this microstrip filter, the optimization process shown in figure 4.19 was used. The original values transformed from the coupled-line filter are parameterized and placed into a single variable/equation element in ADS. A copy of the original variable/equation element is optimized using the Random optimization algorithm, which is very fast and is good at a high-level optimization to get a coarse refinement of the parameters. The most promising permutations from this process are saved as a “seed” for the second optimization step. Each seed is then run through the Gradient

4.3. SIMULATION AND OPTIMIZATION

optimizer which can perform a much finer optimization but also requires more computation time.

It is important to note that neither optimization algorithm is deterministic and thus certain iterations can run into dead-ends where no further useful optimization is possible, while other iterations will produce valuable parameters.

Table 4.3: Optimization algorithms [6]

Algorithm	Description
Random	The simplest of the optimization algorithms. Random values are generated within each parameter's optimization limits and the filter's error function is evaluated to see if an improvement has been achieved. It will successively hone in on parameters that produce the smallest error function.
Gradient	In this algorithm the mathematical gradient of each parameter is evaluated in each iteration and the parameters are adjusted in the direction that slopes toward a lower error function. This algorithm excels at finding local minimums and the parameters obtained are robust against minor variations.

Fortunately due to the symmetrical nature of the hairpin filter, the total number of variables can be reduced by a factor of 2. The parameters used for optimization are shown in table 4.4.

4.3. SIMULATION AND OPTIMIZATION

Table 4.4: Optimization parameters

Parameter	Description
w_{50}	Width of all tracks in the filter, optimally 50Ω .
$u_spacing$	Spacing between legs of a single hairpin, common to all hairpins.
$stub_length$	Length of the top portion of the outermost legs, often used in experimental tuning applications where the stubs can be continuously cut to shift the frequency band.
$outer_gap$	Spacing between the two outermost coupled lines.
$inner_gap$	Spacing between the two innermost coupled lines.
l_{14}	Length of outermost coupled lines.
l_{23}	Length of innermost coupled lines.
tap_length	Length t of the tap, as shown in 4.18. This primarily controls the matching to 50Ω .

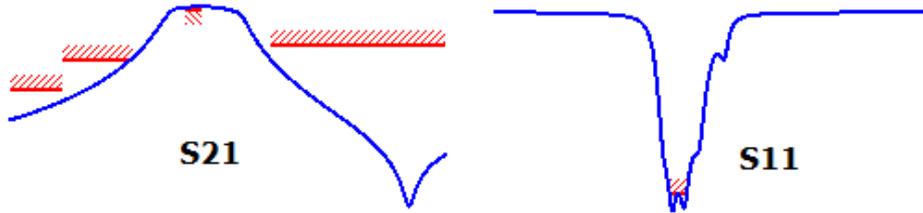


Figure 4.20: Optimization goals

The goals set for the optimization as pictured in figure 4.20 were:

Goal	Range	Reason
$s_{21} > -4$ dB	$1400 \text{ MHz} < f < 1440 \text{ MHz}$	Pass-band through
$s_{21} < -30$ dB	$1120 \text{ MHz} < f < 1280 \text{ MHz}$	1st IF reject
$s_{21} < -50$ dB	$1000 \text{ MHz} < f < 1120 \text{ MHz}$	2nd IF reject
$s_{21} < -20$ dB	$1620 \text{ MHz} < f < 2000 \text{ MHz}$	Upper-frequencies reject
$s_{11} < -30$ dB	$1400 \text{ MHz} < f < 1440 \text{ MHz}$	Return loss minimize

4.3.2 ADS Momentum Simulation

Keysight's ADS provides not only a basic simulation that can be run on optimization, it also provides a Momentum simulator that incorporates all of the EM properties of the circuit board's dimensions and substrate.

4.3. SIMULATION AND OPTIMIZATION

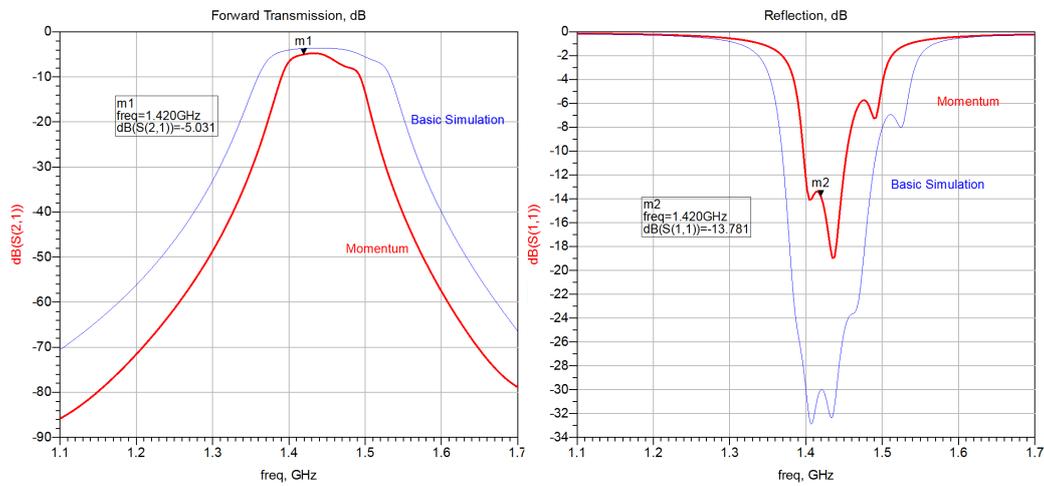


Figure 4.21: Comparison of basic and Momentum simulations in Keysight ADS

As can be expected, when a full EM simulation is run, the performance drops because realistic substrate losses are taken into account. In s_{21} the performance is arguably a bit better, as the primary goal of the preselect filter is to reject the image bands. The Momentum simulation shows a 10-15 dB increase in rejection, with only a 1.3 dB drop in the passband.

4.3.3 CST Frequency Simulation

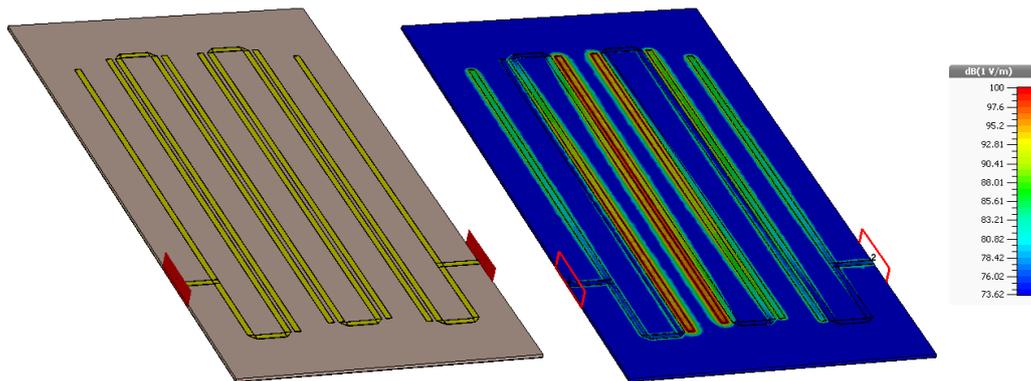


Figure 4.22: Imported design in CST and E-field (absolute) simulation

Once the design had been verified in Keysight ADS's basic and momentum simulations, it was exported as a .DXF (Drawing Exchange Format) file and imported in CST Microwave Studio. The substrate and ground plane

4.3. SIMULATION AND OPTIMIZATION

were added with appropriate material and physical dimensions for the Rogers 4003c substrate to be used. The s-parameters were simulated using CST's Frequency Solver, the results of which are plotted in figure 4.23.

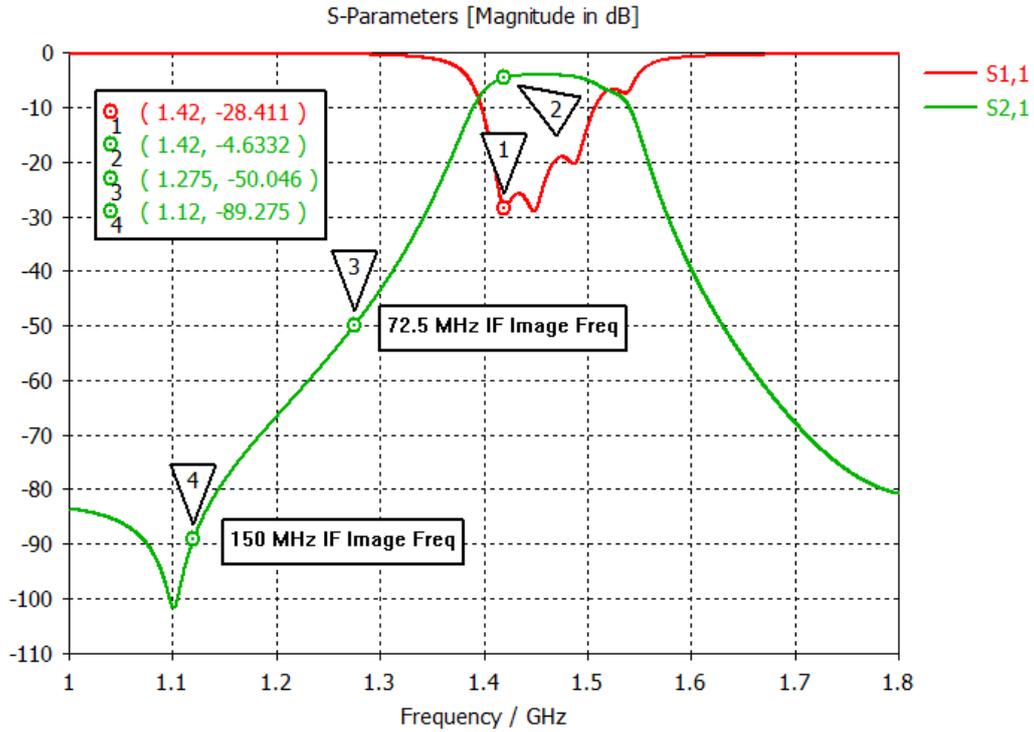


Figure 4.23: Simulation results in CST

The results closely match the results from the ADS momentum simulation.

4.4. MEASUREMENT RESULTS

4.4 Measurement Results

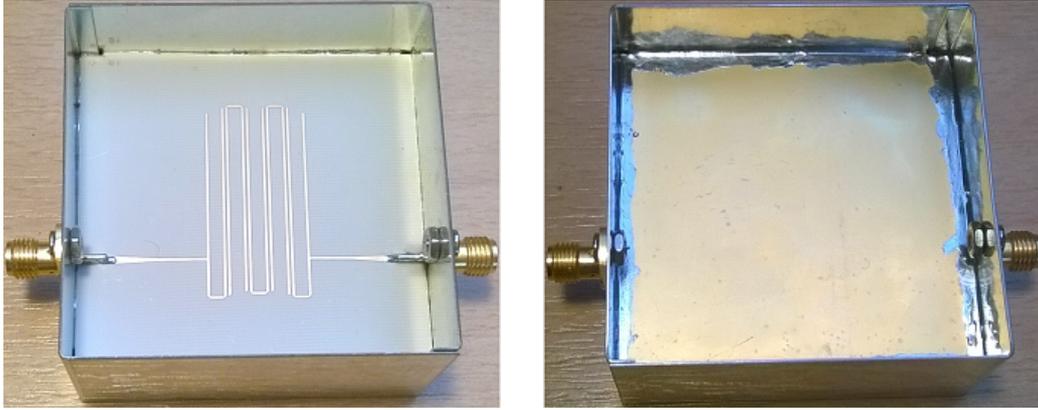


Figure 4.24: The finished filter with enclosure

After the filter's characteristics had been independently verified by two different EM simulators, it was exported as gerber files and sent out for manufacturing at Contag AG. The filter was then soldered into a tin enclosure and attached with SMA connectors and measured with a vector network analyzer. The results are shown in figure 4.25

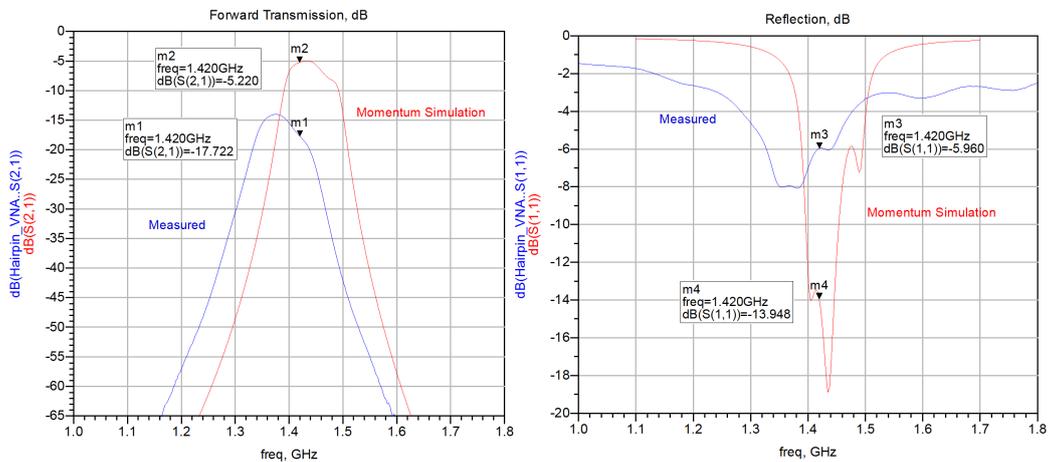


Figure 4.25: Simulated vs. measured results

Unfortunately, as can be seen above, the actual filter's performance varies significantly and unacceptably from the simulated values. There is an obvious frequency shift towards the lower frequencies by about 50 MHz and its best s_{21} is an abysmal -14 dB, showing a 10 dB drop from the simulation.

4.4. MEASUREMENT RESULTS

Generally in printed microstrip filters, frequency shifts are caused by an unexpected or out of tolerance value for the relative permittivity ϵ_r of the substrate. Using ADS's tuning feature the ϵ_r of the simulated substrate was experimentally varied and the value required to create such a shift would be about 3.8. This is quite a difference from the quoted 3.38 parameter by the Rogers Corporation. It is unlikely that such a large variation could occur and the problem more than likely lies on the designer's shoulders.

The dimensions of the hairpin filter were verified with a microscope capable of distance measurements and they were found to be within normal parameters. Additionally the filter was tested without the enclosure, by soldering the SMA connectors directly to the PCB and soldering a small copper strip under the connector to make the ground connection. The difference was minimal.

After troubleshooting proved to be unfruitful, a backup was provided by the Feuerstein Observatory in the form of a filtered LNA with 27 dB of amplification and about 20 MHz bandwidth centered around 1.42 GHz. The filtered LNA was measured using the persistence mode of a spectrum analyzer with a -30 dBm single tone signal as input, as shown in figure 4.26.

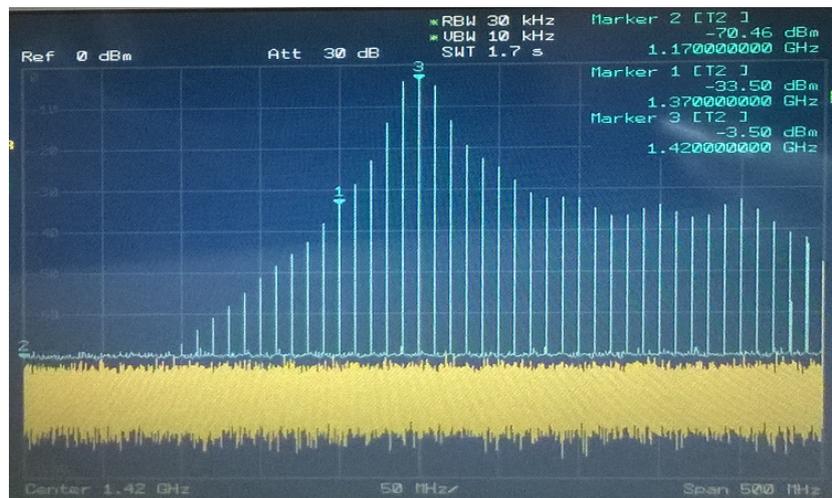


Figure 4.26: Filtered LNA replacement

5.1 Overview

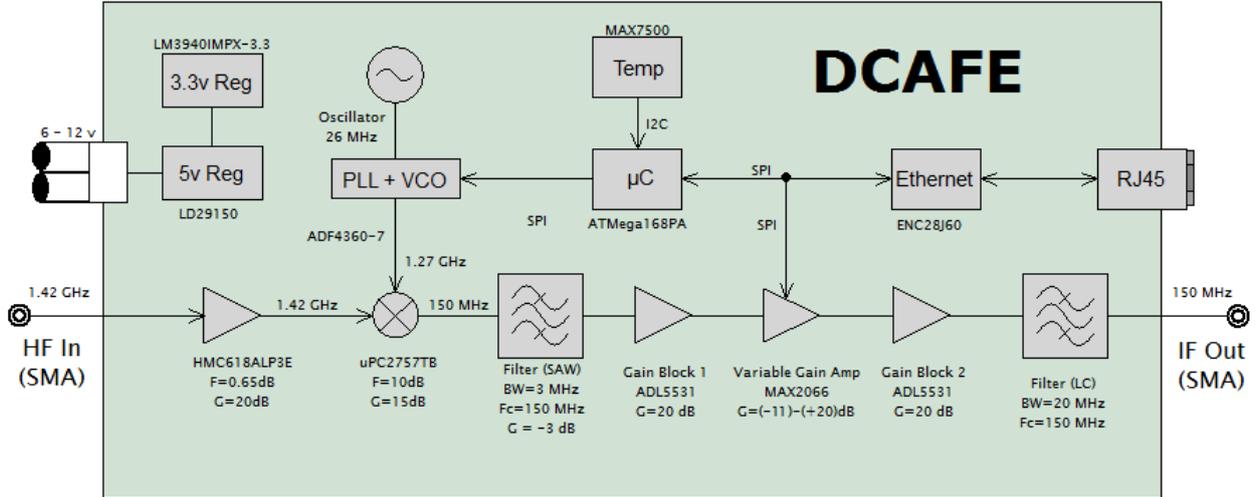


Figure 5.2: The Analog Front-End block diagram

The Digitally Controlled Analog Front-End, or DCAFE for short, is capable of 57-88 dB of variable amplification (factor of 500,000 - 630,000,000) and can perform a down-conversion of 4 MHz of spectrum from 1200-1600 MHz to an intermediate frequency of 150 MHz or 72.5 MHz, depending on its configuration. It is equipped with an on-board temperature sensor so that the temperature can be logged during astronomical measurements and the board is fully controllable over LAN from a computer running a Python script.

When a signal is applied to the HF-In input, it is first amplified by a very low-noise amplifier by 20 dB. It then is converted to a lower frequency by the mixer and local-oscillator combination: $f_{IF} = f_{IN} - f_{LO}$. The local oscillator is a phased-locked loop, or PLL, with an integrated voltage controlled oscillator, or VCO. This combination takes an incredibly accurate and stable 26 MHz reference signal and can output a stable high-frequency signal whose frequency can be varied over a range of several hundred megahertz.

After down-conversion to 150 MHz, a SAW filter removes all frequency components not within a narrow 4 MHz window. Everything within this bandwidth is sent through 3 stages of cascaded amplification, the middle amplifier being a variable gain amplifier, or VGA, that can be varied from -11 to 20 dBs of amplification. The signal is then filtered once more before it is sent along its way to the back-end of the system to be digitized.

5.1. OVERVIEW

The schematics and PCB layout can be found in appendices [A](#) and [B](#).

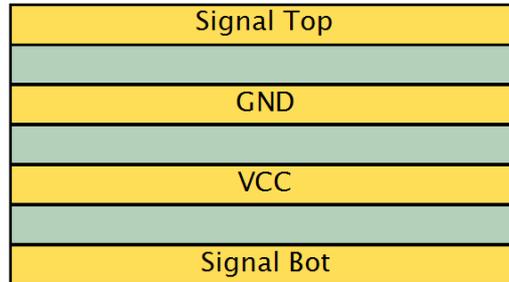


Figure 5.3: Layer stack-up of the FR4 PCB

The PCB was designed as a 4 layer board with an FR4 substrate. The stack-up is shown in figure [5.3](#). The top layer was for all high-frequency signal routing and in digital parts of the board for other routing. The layer below this was an unpartitioned ground plane, which is necessary for controlled impedance microstrip lines. All high-frequency signal paths were routed with a standard $50\ \Omega$ microstrip. The third layer is the voltage supply layer, or VCC, that provides 3.3v and 5v supplies for the individual modules. The last layer is on the very bottom and primarily included only the SPI routing to the PLL/VCO synthesizer and VGA.

All components, even leadless QFN packages, were hand-soldered, saving hundreds of euros on automated placement.

5.2 Mechanical Design

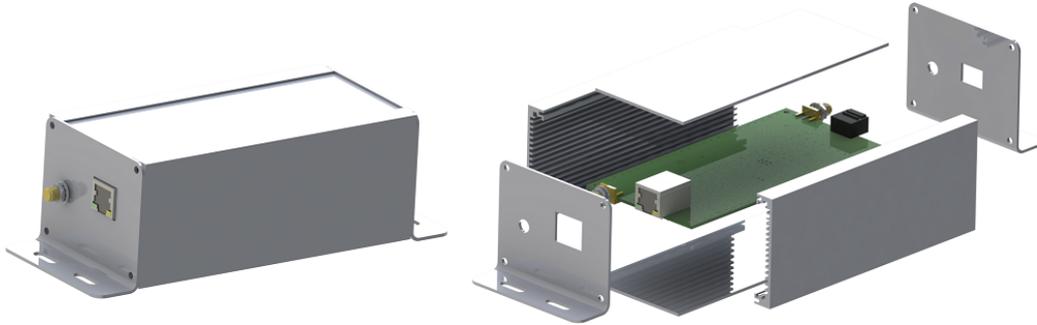


Figure 5.4: Render of the mechanical enclosure

After the PCB was finished, a VRML (Virtual Reality Modeling Language) file was exported from the PCB layout software and imported into a mechanical CAD package. An inexpensive aluminum enclosure designed to fit Eurocards (100 mm x 160 mm) was obtained and modeled in the software using caliper measurements, as no 3D model was available. From there in software the sides of the enclosure could be cut in length (140 mm) and the top and bottom plates cut in width and length (57 mm x 140 mm) to fit the PCB. The ends covers were then designed from scratch with cutouts for connectors and mounting holes, then manufactured in the Institute of Microwaves and Photonics' machine shop.

The enclosure's primary purposes are to provide EMI shielding, make it mountable, and to protect the PCB. In order to accomplish the EMI shielding, conductive rails were designed on along the sides of the bottom of the PCB that make direct contact with the metal enclosure. It also makes ground contact at the two SMA connectors on both ends of the board.

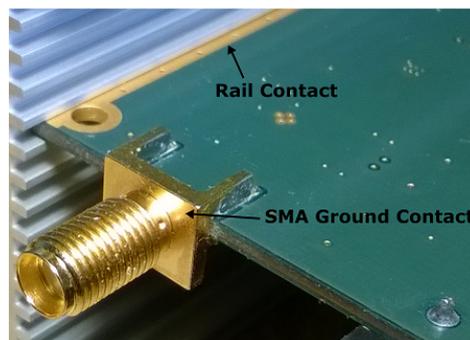


Figure 5.5: Enclosure ground connections

5.3 Firmware

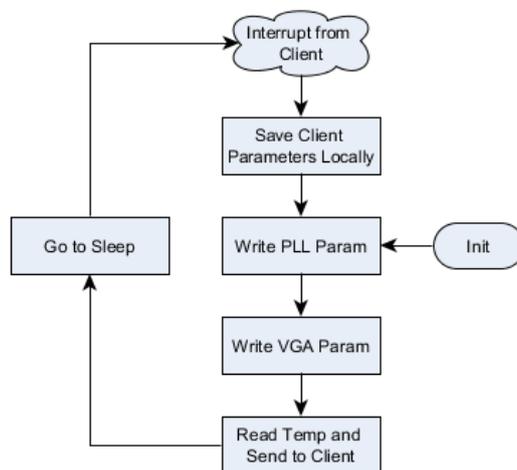


Figure 5.6: Firmware flow chart

The microcontroller is an ATmega168PA and its firmware was written in C. The firmware was designed to be as simple as possible for the end-user to maintain and update. When it first boots, it initializes all of the internal settings and registers then enters into the main-loop. It then writes all of the PLL relevant parameters to the PLL, and the VGA attenuation setting to the VGA. Following this it logs the temperature from the temperature sensor and sends it over the ethernet connection to the client-side computer in the observatory.

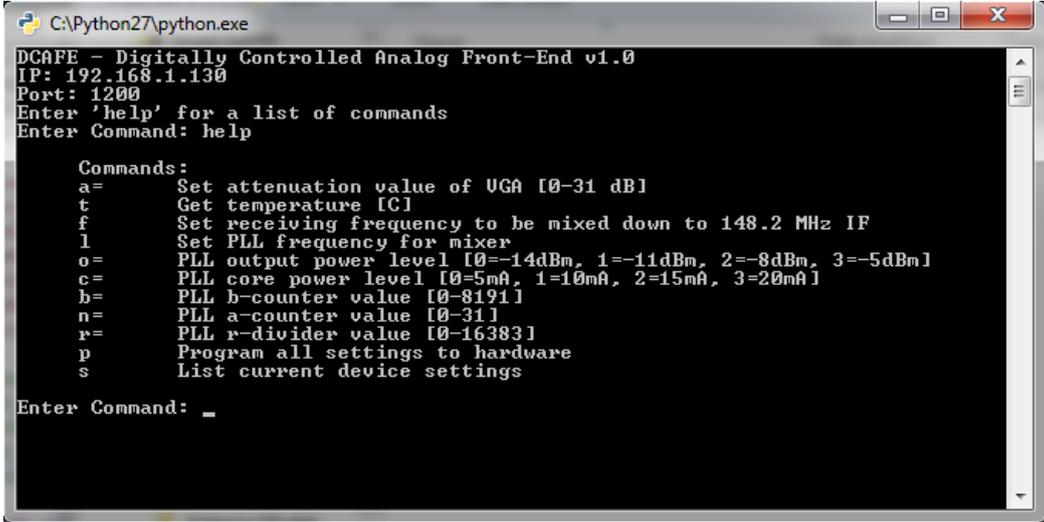
At this point the microcontroller enters sleep-mode, shutting down all digital circuitry on the PCB, making it ideal for low-noise analog measurements, which is highly advantageous for a radio telescope.

When the client makes a request, such as changing the parameters of the PLL, VGA, or sensing the temperature, the microcontroller is awoken from its sleep state by an interrupt from the ethernet controller. The microcontroller can then receive the client's commands and update its local parameter values, which will then be written to the external devices and the process continues.

The data transfer is performed using the User Datagram Protocol, or UDP. The microcontroller parses the incoming UDP packet for the matching password. If the password matches, it will then read the command and the appropriate value if there is one.

Additionally, the ATmega is configured with a watch-dog timer to reset the device should it crash or run into an infinite loop. The top-level code of the firmware can be found in appendix C.

5.4 Control Software

A screenshot of a Windows command prompt window titled "C:\Python27\python.exe". The window displays the output of a Python program named "DCAPFE - Digitally Controlled Analog Front-End v1.0". The program shows its IP address as 192.168.1.130 and its port as 1200. It prompts the user to enter a command, and the user has entered "help". The program then displays a list of commands and their descriptions: "a=" for setting attenuation, "t" for getting temperature, "f" for setting receiving frequency, "l" for setting PLL frequency, "o=" for setting PLL output power level, "c=" for setting PLL core power level, "b=" for setting PLL b-counter value, "n=" for setting PLL a-counter value, "r=" for setting PLL r-divider value, "p" for programming settings to hardware, and "s" for listing current device settings. The prompt "Enter Command: _" is visible at the bottom.

```
C:\Python27\python.exe
DCAPFE - Digitally Controlled Analog Front-End v1.0
IP: 192.168.1.130
Port: 1200
Enter 'help' for a list of commands
Enter Command: help

Commands:
a=      Set attenuation value of UGA [0-31 dB]
t       Get temperature [C]
f       Set receiving frequency to be mixed down to 148.2 MHz IF
l       Set PLL frequency for mixer
o=     PLL output power level [0=-14dBm, 1=-11dBm, 2=-8dBm, 3=-5dBm]
c=     PLL core power level [0=5mA, 1=10mA, 2=15mA, 3=20mA]
b=     PLL b-counter value [0-8191]
n=     PLL a-counter value [0-31]
r=     PLL r-divider value [0-16383]
p      Program all settings to hardware
s      List current device settings

Enter Command: _
```

Figure 5.7: Control software written in Python

In order to configure the analog front-end over the network through its ethernet interface, a short program in Python was developed. The software simply takes commands from the user and packages them into the appropriate UDP packet format and sends them. The firmware on the ATmega microcontroller will parse the packet data, verify that the values are within the allowed ranges, and store the values locally. All aspects of the PLL synthesizer, variable attenuator and temperature sensor can be configured or when applicable queried.

The Python code for this program can be found in appendix D.

5.5 Measurements

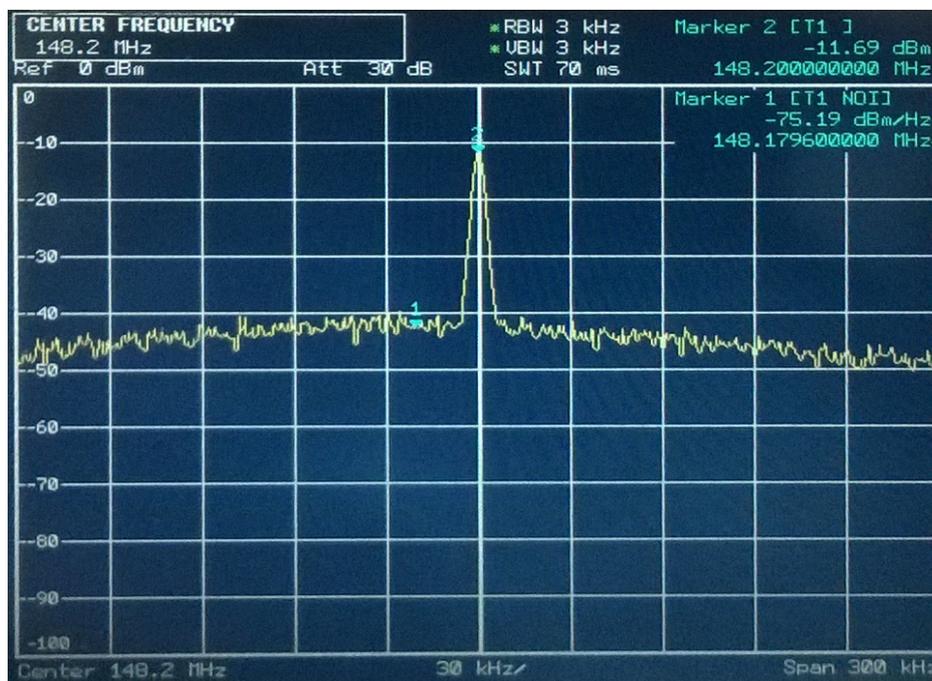


Figure 5.8: Output spectrum with -100 dBm input @ 1.42 GHz

To measure the analog front-end, a -100 dBm sine-wave was generated at a frequency of 1.42 GHz and fed into the input of the PCB. The output was measured with a spectrum analyzer, the results of which can be seen in figure 5.8. The attenuation of the front-end was set to zero, which means that it is capable of 88 dB of amplification.

The noise-factor of the device was determined using the gain method, which can be calculated as follows: [7]

$$F = P_{NOUT} + 174 \text{ dBm/Hz} - \text{Gain}$$

Where P_{NOUT} is the output noise density, 174 dBm/Hz is the ambient noise density at room temperature, and Gain is the amplification of the system. Inserting values for these factors:

$$F = -75 \text{ dBm/Hz} + 174 \text{ dBm/Hz} - 88 \text{ dB} = 11 \text{ dB}$$

Thus, the noise factor of the analog front-end is 11 dB, which is well within the range of an average receiver.

Chapter 6

System Measurements



6.1 Putting It All Together

After all modules of the system had been tested both individually and cascaded in a lab environment, it was time to start mounting the equipment and routing all of the necessary coaxial and ethernet connections. For the coaxial cable, H-155 was chosen due to its low losses, moisture resistance, and ease of installation. All connections were either terminated in SMA or N-type connectors depending on which connector the device required. Generic CAT-V cable was chosen for the ethernet connection.

Luckily, the members of the Feuerstein Observatory had spent a significant amount of time building a frame for the parabolic reflector to sit on that is motorized in two axes and could be controlled through the LAN by software. With that, it is possible to give extremely precise coordinates to direct the antenna, including compensation for the earth's rotation.

6.2. CALIBRATION

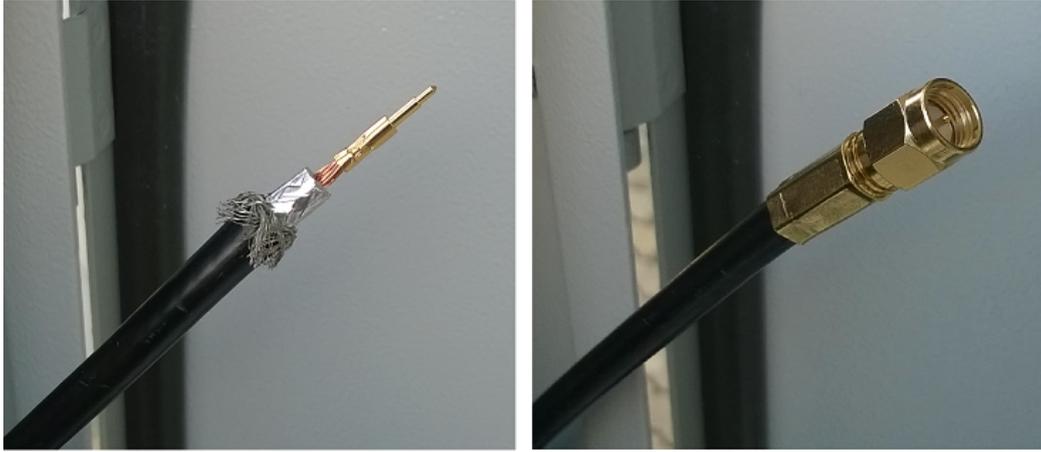


Figure 6.1: Routing and crimping H-155 cable with SMA connection

Two high quality, metal control enclosures from Rittal were procured for the front- and back-end equipment to be mounted. Holes were drilled in the enclosures and cable through-ports were added for the connections and the equipment was mounted on an aluminum plate after threaded holes were drilled.

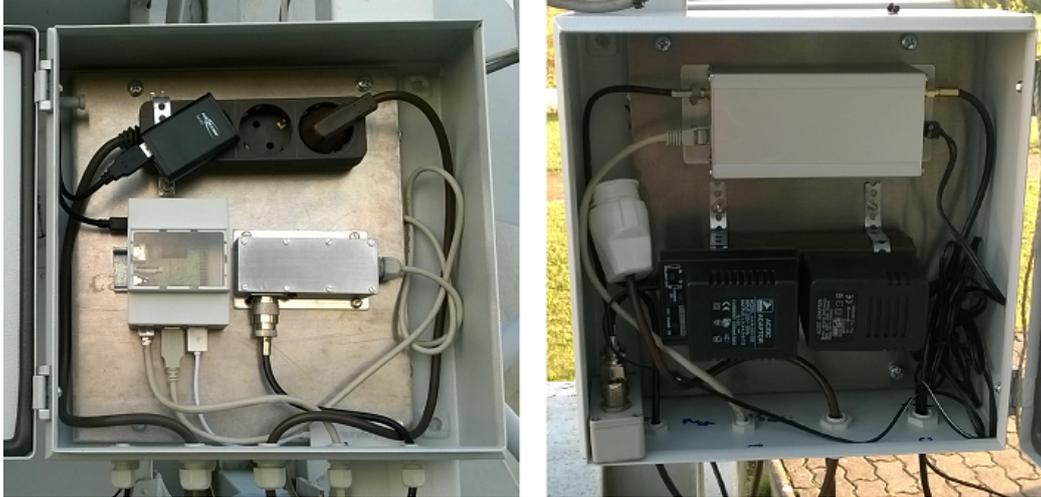


Figure 6.2: Back- and front- ends in water and dust resistant cabinets

6.2 Calibration

For this system two calibration processes were conceived to calibrate both amplitude and frequency measurements. Due to the inhomogeneous ampli-

6.2. CALIBRATION

fication and attenuation of different components in the chain, most notably the SAW filter, there is a ripple across the 4 MHz of the received bandwidth, as shown in figure 6.3.

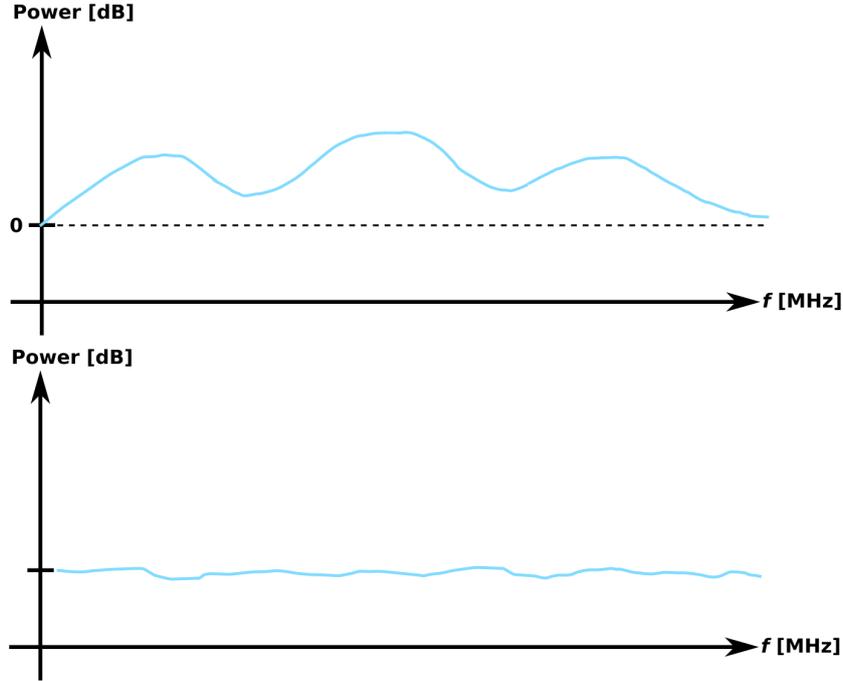


Figure 6.3: Calibration procedure showing pre- and post-calibration spectrum

In order to compensate for this ripple and other system noise to make the spectrum flat, the input connection from the antenna is automatically switched to a $50\ \Omega$ resistor using a “Dicke Switch”. The output spectrum is then measured, averaged, and saved as S_{offset} for later use. The Dicke Switch can then be switched back to the antenna where the dish will be pointed at the warm ground or tree line where only white-noise is measured. This spectrum is then also averaged and recorded as S_{white} . In order to produce the calibrated spectrum so that very precise measurements can be made, all of the spectrums are translated into linear values and adjusted by:

$$S_{calibrated} = \frac{S_{measured} - S_{offset}}{S_{white} - S_{offset}}$$

To calibrate the frequency, an object with a known relative velocity to earth is measured, the measured frequency shift compared to the actual, and the spectrum shifted accordingly.

6.3 Milky Way Crossing

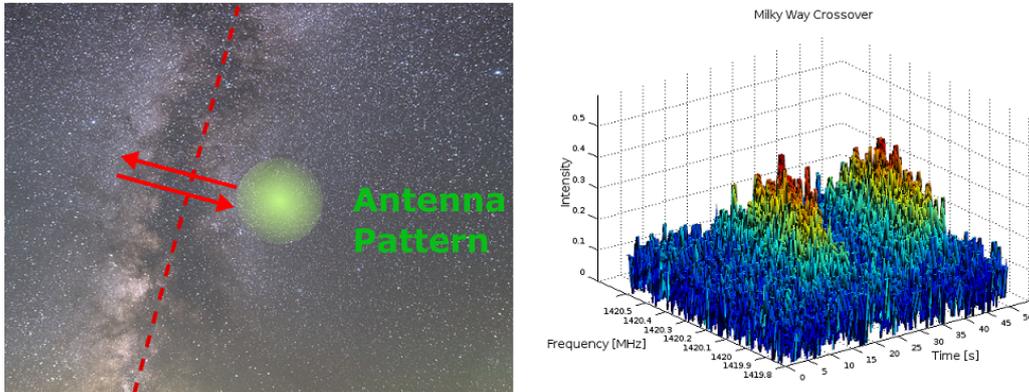


Figure 6.4: Back and forth crossing of the Milky Way

For an initial test, the newly designed radio telescope was driven across a strip of the Milky Way, then a few seconds later back again, as depicted in figure 6.4. This produced the two peaks shown in the spectrum. The symmetry of the signals as shown in figure 6.5 proves that the signal is from neutral hydrogen radiation and not just random noise.

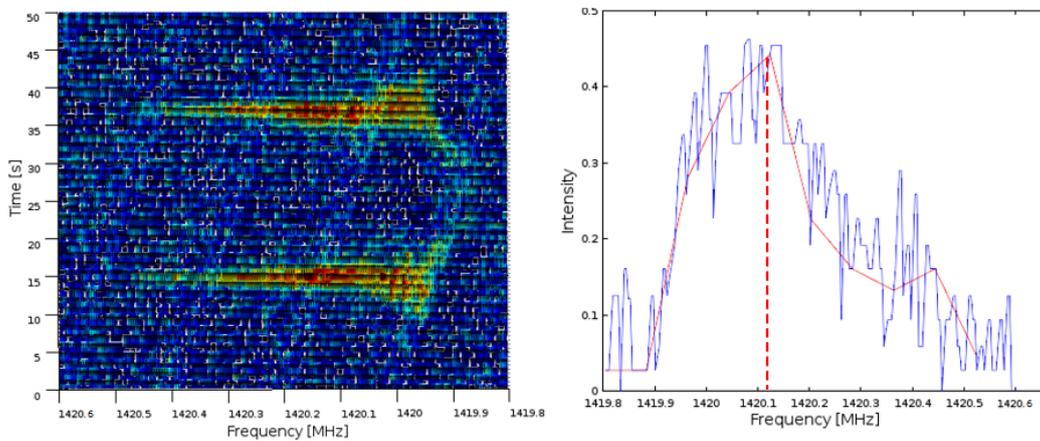


Figure 6.5: 2D data

When the data is averaged and the peak is measured, a frequency displacement of approximately 280 kHz is evident. Using the rearranged doppler equation from section 1.2 the relative velocity of the hydrogen cloud can be calculated:

6.3. MILKY WAY CROSSING

$$v = \frac{\Delta f}{f_r} c_0 = \frac{-280 \text{ kHz}}{1420.406 \text{ MHz}} \cdot 300e6 \frac{m}{s} \approx -60 \frac{\text{km}}{\text{s}}$$

Therefore, the hydrogen in the narrow strip measured is moving away from Earth with an approximate velocity of 60 km/s.

It is important to note that this was a very quick measurement meant as a demonstration of the radio telescope's functionality. For high quality scientific data the measurement will require minutes or even hours of averaging the received signal to reduce noise.

Chapter 7

Summary and Looking Forward



As a part of a bachelor's thesis at Friedrich-Alexander University Erlangen-Nürnberg and in cooperation with the Feuerstein Observatory and the FAU Institute for Microwaves and Photonics, a radio telescope for hydrogen-line astronomy was designed, constructed, and tested. Three parts of the system, the antenna, the microstrip filter, and the analog front-end were developed completely from scratch, with everything but the filter proving to be total successes.

The analog section of the radio telescope is now completely finished, however, there is still a lot of work that needs to be done by the Feuerstein Observatory crew on the software to capture and process incoming data. In time, this radio telescope promises to be an invaluable scientific instrument to support the Feuerstein Observatory's mission of bringing scientific engagement to the public.

There is no reason that this system is limited to astronomy. The technology used is capable of receiving virtually any signal in the 1200 - 1600 MHz band, simply by changing out the preselect filter for the appropriate use. One proposed application is using the system to track satellites in polar orbits flying overhead, capturing, digitizing and processing their transmissions.

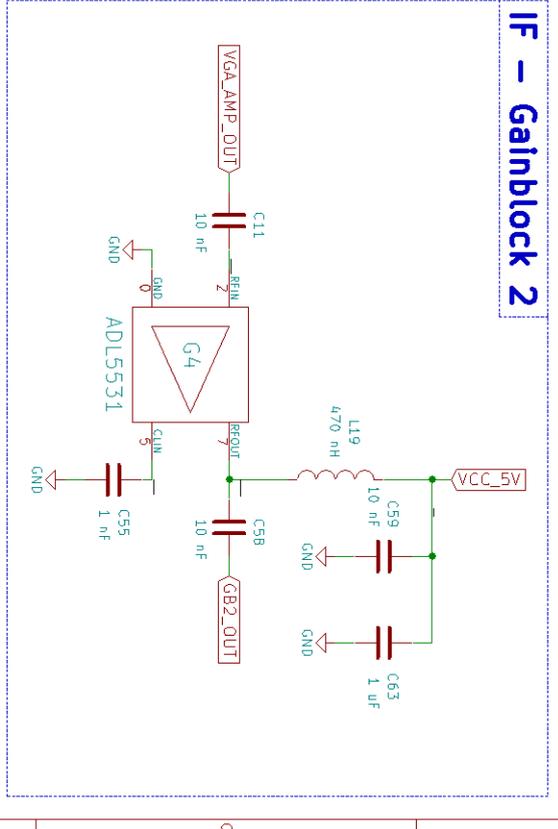
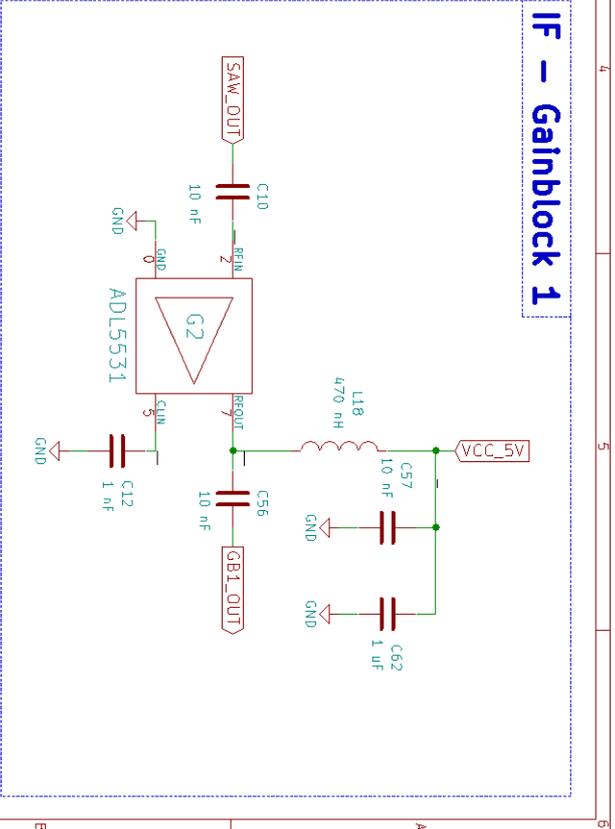
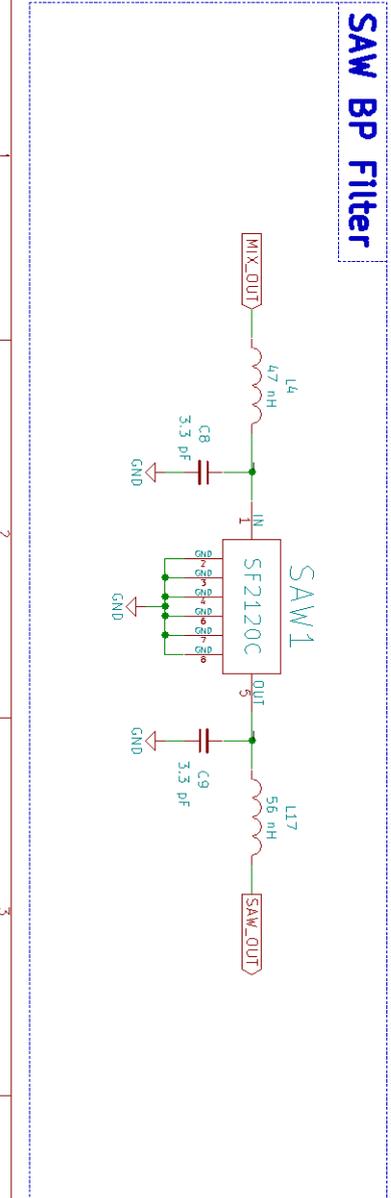
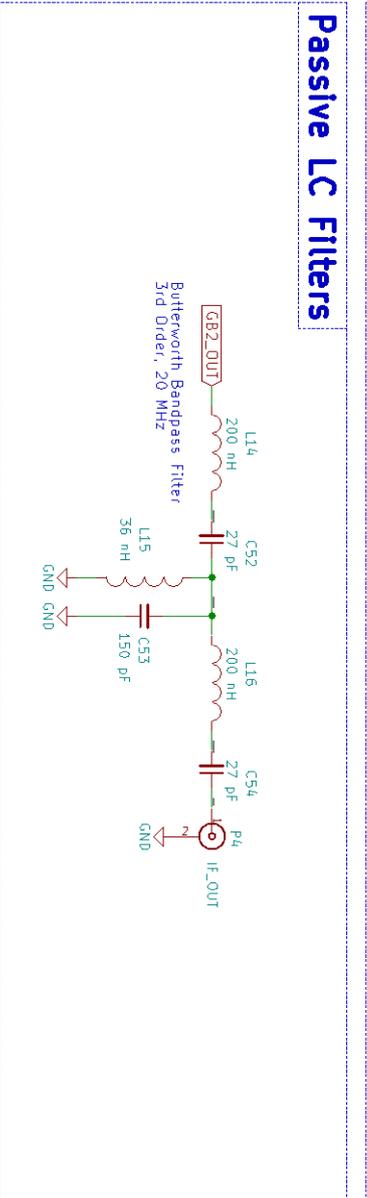
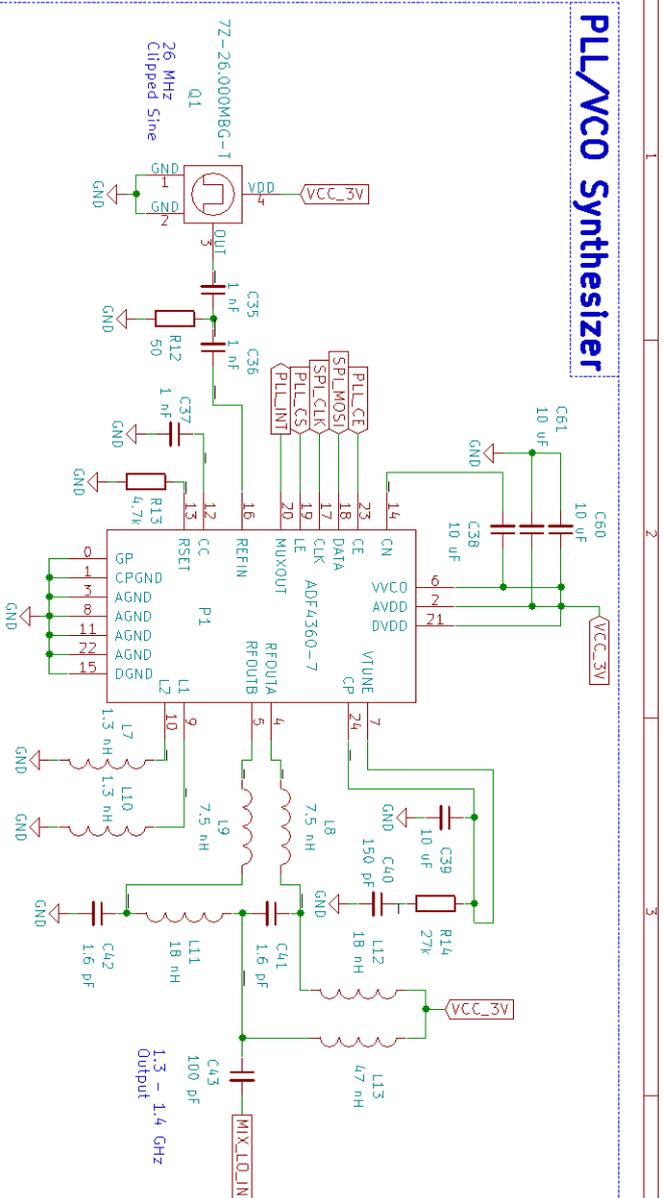
7.1 Special Thanks

- Michael Gottinger: Adviser
- Martin Vossiek: Senior Adviser
- Frank Fleischmann, Alfred Schmiedl and the Sternwarte Feuerstein team
- Ottmar Wick
- Johannes Ringel
- Jürgen Popp and Günter Bauer of the LHFT Machine Shop
- Emanuel Sizmann

Appendices

Appendix A

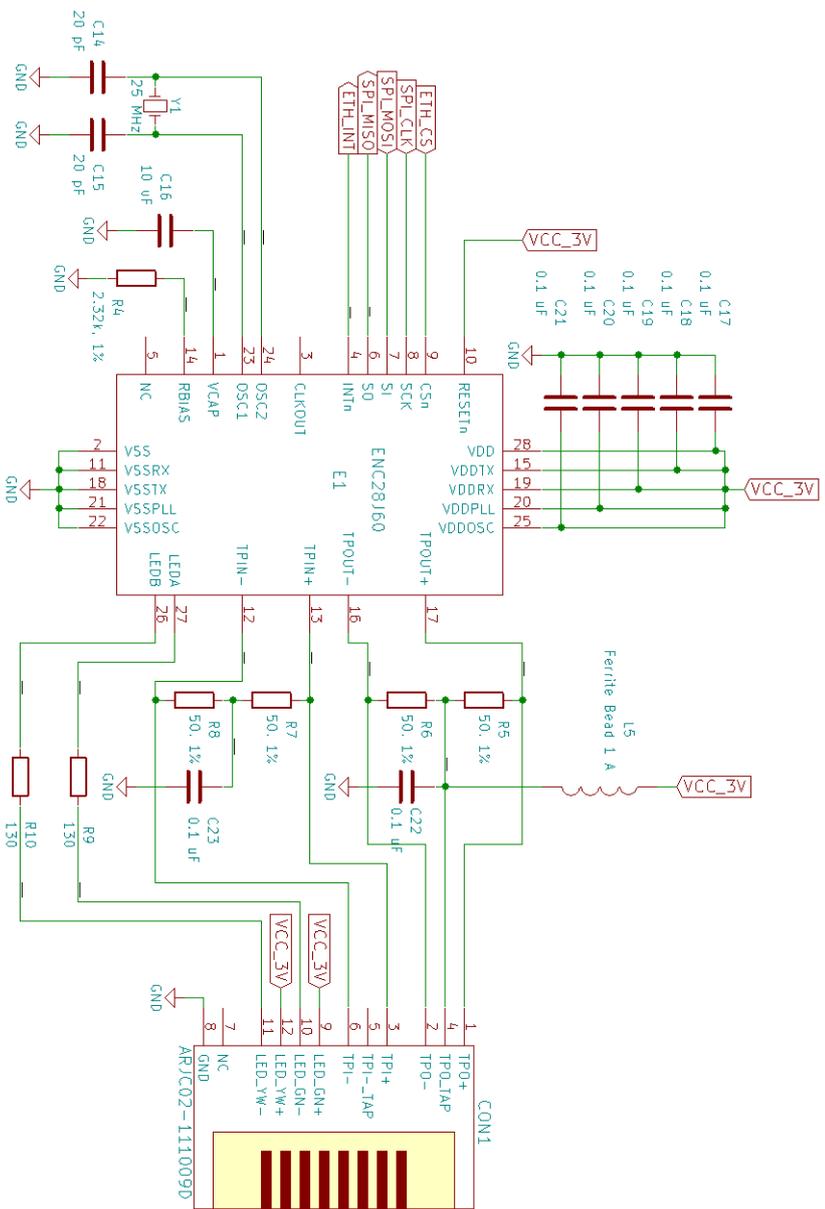
Analog Front-End Schematics



DCAFÉ – Digitally Controlled Analog Front-End

Sheet: 7/16_Layer2/2
 File: HF_Layout2.sch
Title:
 Size: A4
 Date: June 2016
 Kicad E.D.A. kicad 4.0.2-stable
 Rev: v1.0
 Id: 4/5

Ethernet Connection



DCAFÉ – Digitally Controlled Analog Front-End

Sheet: /Digital/Layout/
File: Digital/Layout.sch

Title:

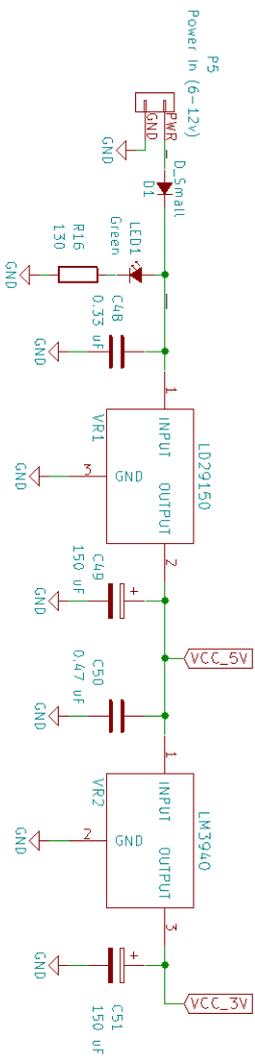
Size: A4 Date: June 2016

KiCad E.D.A. KiCad 4.0.2-stable

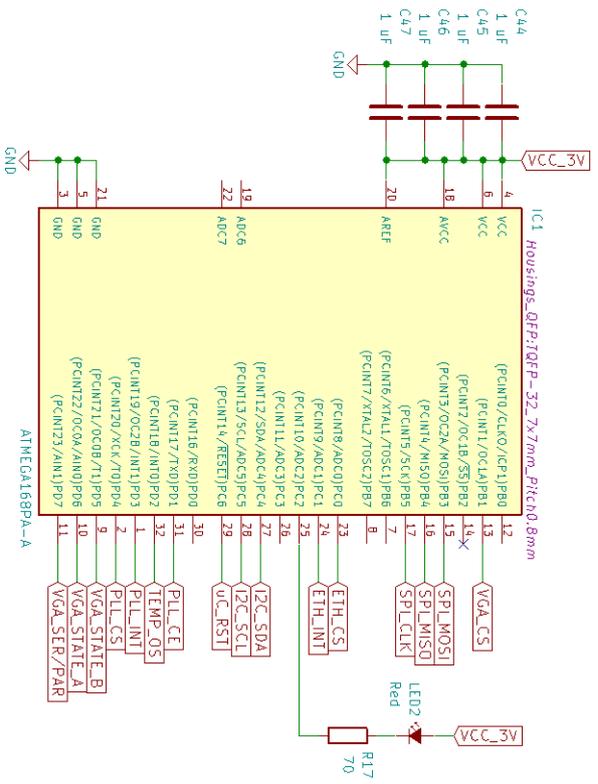
Rev: v1.0

Id: 3/5

Voltage Regulation



Microcontroller



Appendix B

Analog Front-End PCB



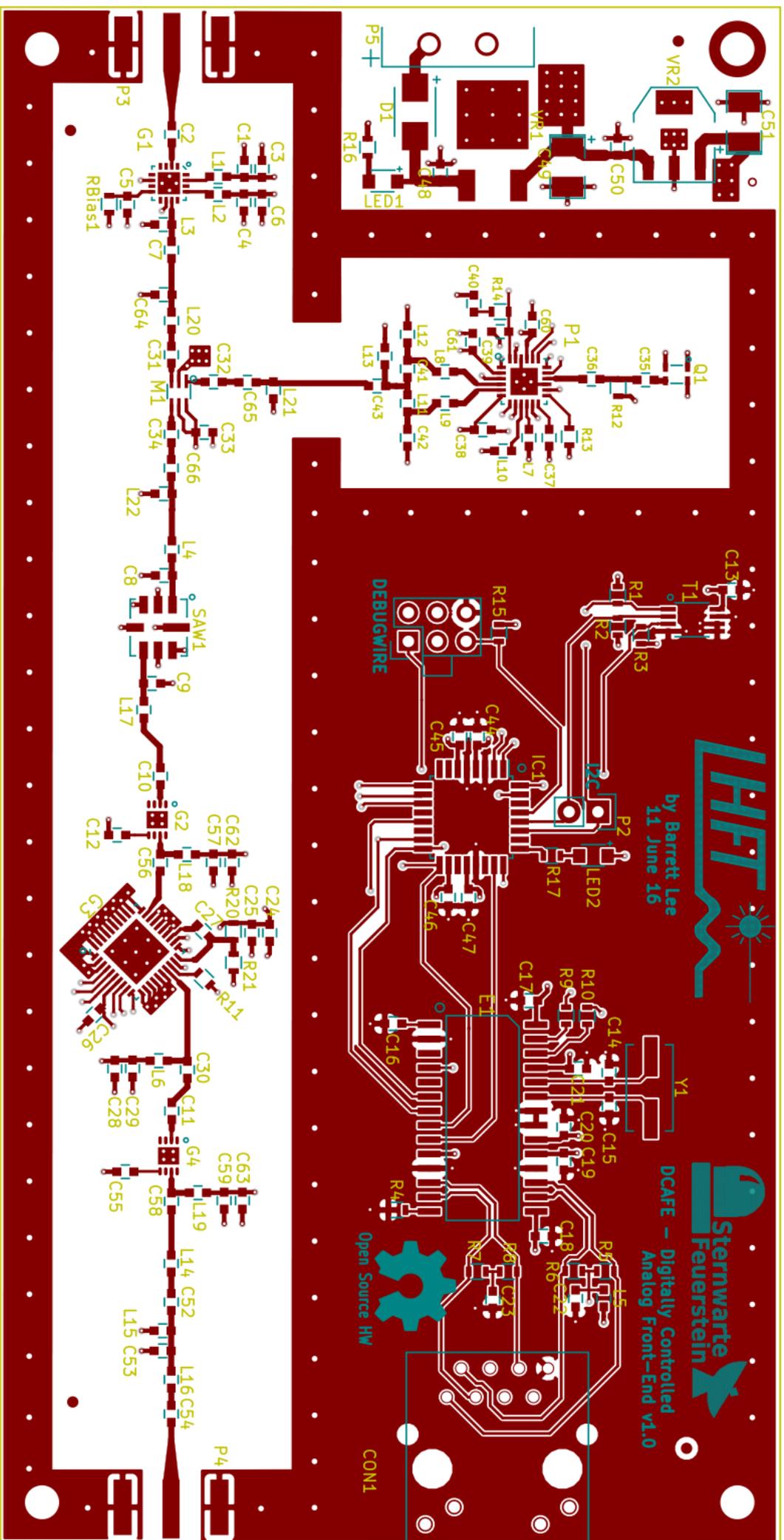
by Barrett Lee
11 June 16

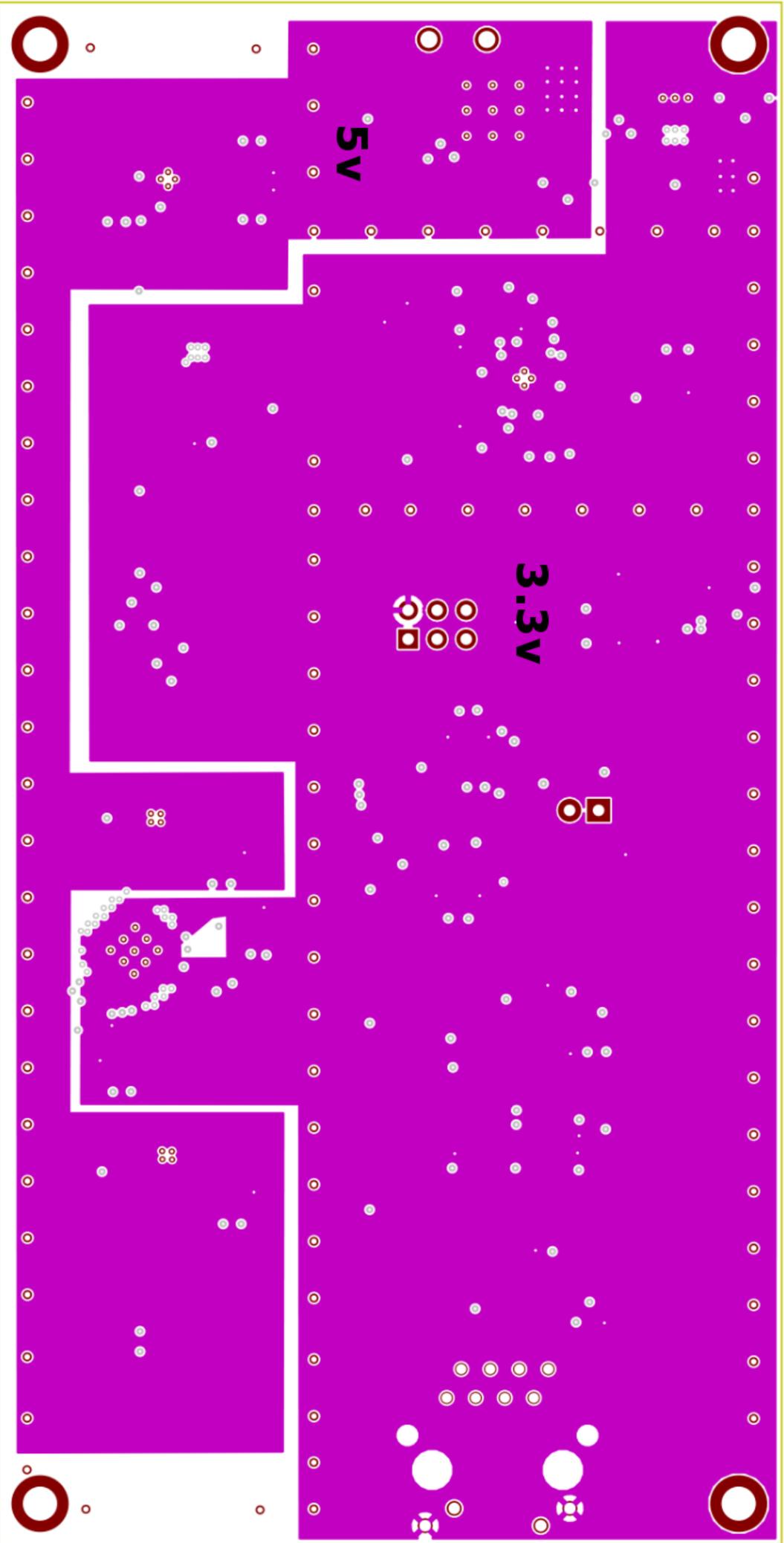


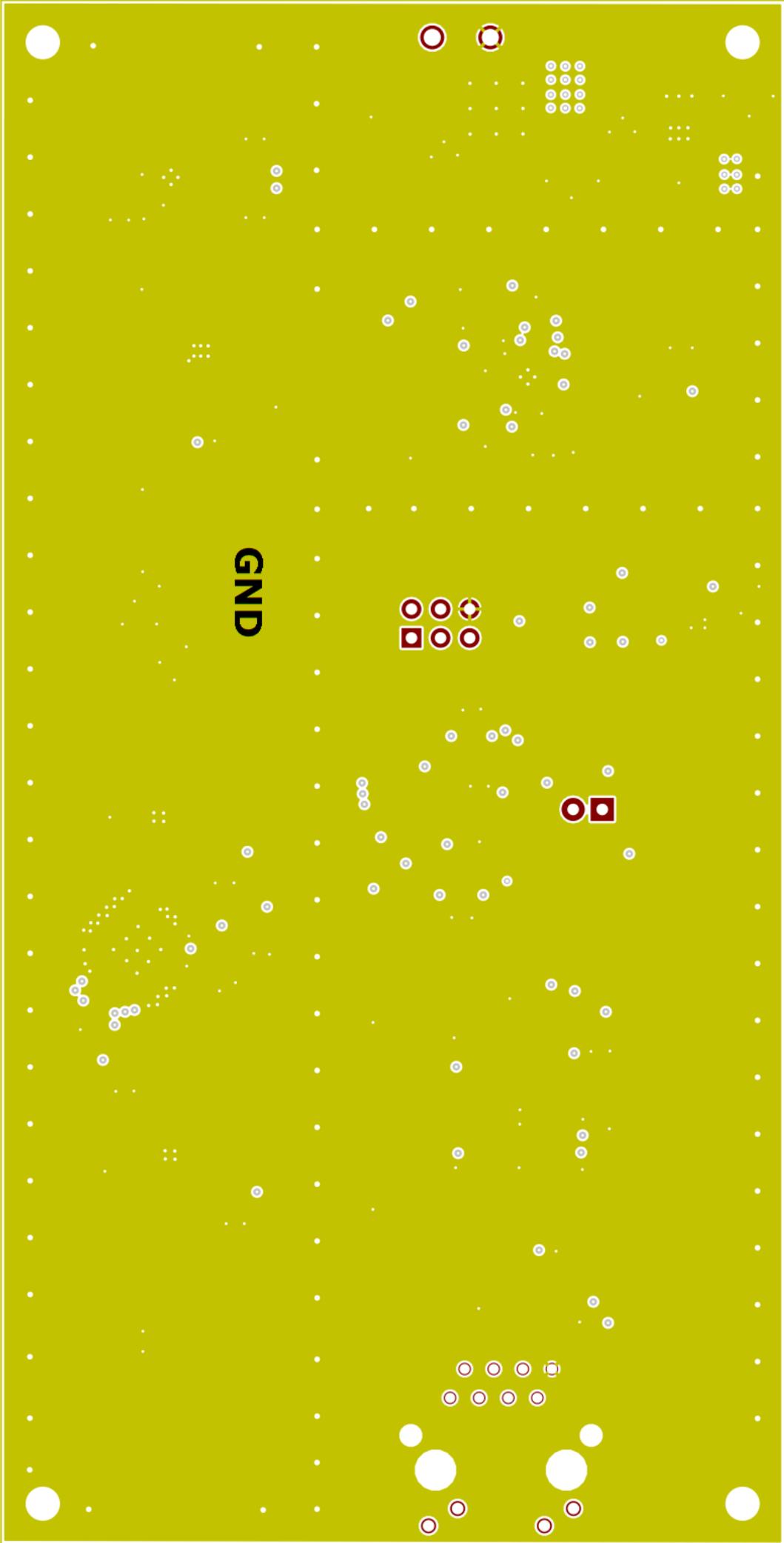
DCAFÉ - Digitally Controlled
Analog Front-End v1.0



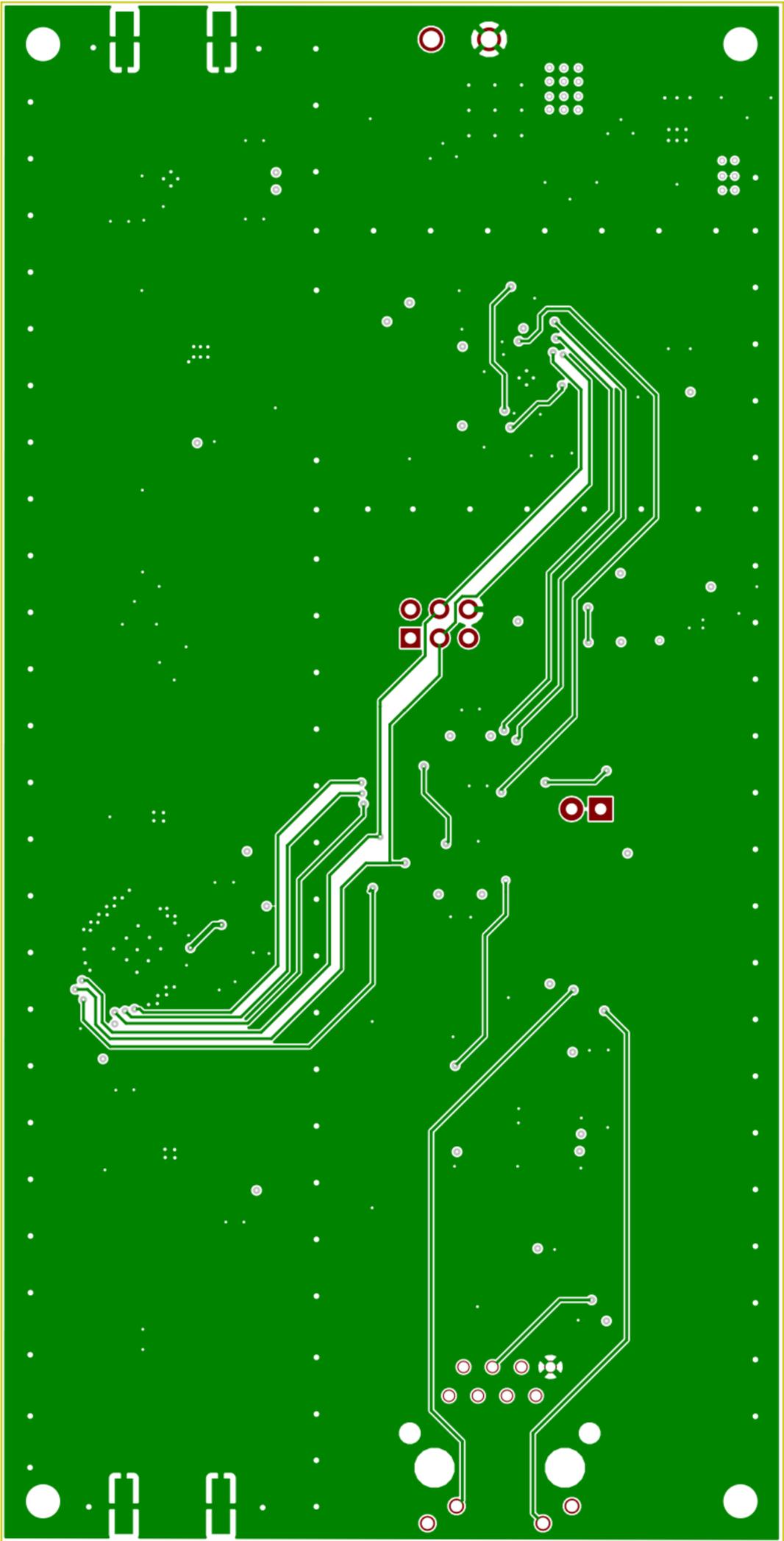
CON1







GND



Appendix C

Analog Front-End Firmware

```
1 // Firmware for the Digitally Controlled Analog Front-End
2 // Author: Barrett Lee
3 // Date: 14 July 2016
4
5 #include "Firmware.h"
6
7 // AVR Includes
8 #include <avr/io.h>
9 #include <util/delay.h>
10 #include <inttypes.h>
11 #include <stdio.h>
12 #include <avr/wdt.h>
13
14 // Bus Includes
15 #include "spi.h"
16 #include "lan.h"
17 #include "i2cmaster.h"
18
19 // Hardware Includes
20 #include "MAX7500_TEMP.h"
21 #include "MAX2066_VGA.h"
22 #include "ADF4360_PLL.h"
23
24 ///////////////////////////////////////////////////////////////////
25 // Variables //
26 ///////////////////////////////////////////////////////////////////
27 // PLL registers
28 PLL_C_REG c_reg;
29 PLL_N_REG n_reg;
30 PLL_R_REG r_reg;
31
32 // Dynamically changeable LAN data
33 LAN_DATA lan_data;
34
35 // Main Loop
36 int main(void)
37 {
38
39     initLED(); // Init debug LED
40     spi_init(); // Init SPI bus
```

```

41 initVGA(); // Init variable gain amplifier
init_PLL(&c_reg, &n_reg, &r_reg); // Init phase-locked loop
synthesizer and registers
43 init_lan(); // Init LAN bus
i2c_init(); // Init i2c bus
45 WDTconfig(); // Configure watch-dog timer

47 offLED();

49 // Init local LAN data
lan_data.attenuation = 0;
51 lan_data.pll_a_counter = n_reg.a_counter;
lan_data.pll_b_counter = n_reg.b_counter;
53 lan_data.pll_core_pwr = c_reg.core_pwr_lvl;
lan_data.pll_output_pwr = c_reg.output_pwr_lvl;
55 lan_data.pll_r_divider = r_reg.r_counter;

57 while (1)
{
59 // Get data from network
61 IO_lan(&lan_data);

63 // If the client requests parameter changes, update hardware
if (lan_data.program_new == 1)
65 {
// Program all hardware settings
67 update_hardware(lan_data);

69 // Reset flag
lan_data.program_new = 0;
71 }

73 // Reset watch-dog timer
wdt_reset();

75 }
77 }

79 // Write all settings to all hardware
void update_hardware(LAN_DATA lan_data)
81 {
onLED();

83 // Write attenuation setting to VGA
85 setVGA_Atten(lan_data.attenuation);

87 // Write PLL settings to PLL
c_reg.output_pwr_lvl = lan_data.pll_output_pwr;

```

```
89  c_reg.core_pwr_lvl = lan_data.pll_core_pwr;
    n_reg.b_counter = lan_data.pll_b_counter;
91  n_reg.a_counter = lan_data.pll_a_counter;
    r_reg.r_counter = lan_data.pll_r_divider;
93
    program_PLL(c_reg, n_reg, r_reg);
95
    offLED();
97 }
99 // Configure watch dog timer
void WDTconfig(void)
101 {
    MCUSR &= ~(1 << WDRF);           // Clear WD Flag
103
    wdt_reset();
105    wdt_enable(WDTO_4S);           // Enable WDT and set timer to 4 s
107 }
```

Appendix D

Python Software for Control of the Analog Front-End

```
1 from __future__ import division
import socket, time
3
4 # Connection Information
5 UDP_IP = "192.168.1.130"
6 UDP_PORT = 1200
7
8 # Open Socket to device
9 sock = socket.socket(socket.AF_INET,
10                      socket.SOCK_DGRAM)
11
12 # Variables
13 attenuation = 30
14
15 print "DCAFE - Digitally Controlled Analog Front-End v1.0"
16 print "IP: %s" % UDP_IP
17 print "Port: %s" % UDP_PORT
18 print "Enter 'help' for a list of commands"
19
20 # Set the frequency of the PLL
21 def set_pll_freq(freq):
22
23     # Prescaler value
24     p = 8
25
26     # Reference Frequency
27     fext = 26e6 # external clock
28     r = 260 # r-divider register
29     fref = fext / r
30
31     # Calc N
32     N = freq / fref
33
34     # Calc b
35     b = int(N / p)
36
37     # Calc a
38     a = int(N - p*b)
```

```

39 # PLL Freq
    freq_pll = int(((p*b)+a)*fref)
41
42 # Send b value
43 MESSAGE = "cmd,b=" + str(b) + ";"
    sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
45 time.sleep(0.5)
    # Receive echo
47 echo = sock.recv(1024)
    print echo
49
50 # Send a value
51 MESSAGE = "cmd,n=" + str(a) + ";"
    sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
53 time.sleep(0.5)
    # Receive echo
55 echo = sock.recv(1024)
    print echo
57
    return freq_pll
59
60 # Set the receiving frequency to be mixed down to 148.2 MHz
61 # Range ~(1200 - 1600 MHz) without filtered LNA
62 # Range ~(1300 - 1600 MHz) with filtered LNA
63 def set_rec_freq(freq):
64
65     # Intermediate frequency to be mixed down to
        f_if = 148.2e6
67
68     # Calculate required PLL frequency
69     f_pll = freq - f_if
        f_pll = set_pll_freq(f_pll)
71
72     print ">> Receiving frequency: %s MHz" % ((f_pll + f_if) / 1
        e6)
73
74 # Main Loop
75 while(1):
76     # Get command from user
77     command = raw_input("Enter Command: ")
78
79     # Help menu
80     if (command == "help"):
81         print " "
82         print "      Commands:"
83         print "      a=      Set attenuation value of VGA [0-31
dB]"
84         print "      t          Get temperature [C]"

```

```

85     print "      f      Set receiving frequency to be mixed
down to 148.2 MHz IF"
86     print "      l      Set PLL frequency for mixer"
87     print "      o=     PLL output power level [0=-14dBm,
1=-11dBm, 2=-8dBm, 3=-5dBm]"
88     print "      c=     PLL core power level [0=5mA, 1=10mA,
2=15mA, 3=20mA]"
89     print "      b=     PLL b-counter value [0-8191]"
90     print "      n=     PLL a-counter value [0-31]"
91     print "      r=     PLL r-divider value [0-16383]"
92     print "      p      Program all settings to hardware"
93     print "      s      List current device settings"
94     print " "

95
96     elif(command == "f"):
97         # Get frequency from user
98         frequency = float(raw_input("Enter frequency to receive
[MHz]: "))
99
100        # Set frequency
101        set_rec_freq( int(frequency * 1e6) )

102
103        elif (command == "l"):
104            # Get frequency from user
105            frequency = float(raw_input("Enter desired PLL frequency
[MHz]: "))
106
107            # Set frequency
108            pll_freq = set_pll_freq(int(frequency * 1e6))
109            print ">> PLL frequency: %s MHz" % (pll_freq / 1e6)

110
111        # Send command to device and receive echo
112        else:
113            # Send payload
114            MESSAGE = "cmd," + command + ";"
115            sock.sendto(MESSAGE, (UDP_IP, UDP_PORT))
116            time.sleep(0.5)
117            # Receive echo
118            echo = sock.recv(1024)
119            print echo

```

Bibliography

- [1] Martin Vossiek, *Skript zur Vorlesung und Übung - Hochfrequenztechnik* Lehrstuhl für Hochfrequenztechnik, WS 2015/16
- [2] Martin Vossiek, *Passive Bauelemente und deren HF-Verhalten* Lehrstuhl für Hochfrequenztechnik, SS 2015
- [3] John D. Kraus, *Antennas for All Applications 3rd Ed.* McGraw-Hill, 2002
- [4] Thomas A. Milligan, *Modern Antenna Design 2nd Ed.* John Wiley & Sons, 2005
- [5] H. Paul Schuch, *SETI League Technical Manual* SETI League <http://www.setileague.org/hardware/feedchok.htm>
- [6] Summary of ADS Optimizers
http://cp.literature.agilent.com/litweb/pdf/ads2008/optstat/ads2008/Summary_of_Optimizers.html
- [7] Three Methods of Noise Figure Measurement *Maxim Integrated*
<https://www.maximintegrated.com/en/app-notes/index.mvp/id/2875>
- [8] Gerhard Merziger, *Formeln + Hilfen Höhere Mathematik 6. Auflage* Bionomi Verlag, 2010
- [9] Horst Kuchling, *Taschenbuch der Physik 19. Auflage* Hanser Verlag, 2007
- [10] David M. Pozar, *Microwave Engineering 4th Ed.* John Wiley & Sons, 2012
- [11] Hong & Lancaster, *Microstrip Filters for RF/Microwave Applications* John Wiley & Sons, 2001
- [12] Hyperphysics, *Georgia State Astrophysics*
<http://hyperphysics.phy-astr.gsu.edu/hbase/quantum/h21.html>